DOCTEUR EN SCIENCES

Développements en phylogénomique: comparaisons de génomes et estimation de grandes phylogénies

Helaers, Raphaël

Award date:
2010

Awarding institution:
Universite de Namur

FACULTÉ DES SCIENCES
DÉPARTEMENT DE BIOLOGIE

DÉVELOPPEMENTS EN PHYLOGÉNOMIQUE : COMPARAISONS DE GÉNOMES ET ESTIMATION DE GRANDES PHYLOGÉNIES

Dissertation présentée par
Raphaël Helaers
en vue de l'obtention du grade
de Docteur en Sciences

Composition du jury :
Prof. Eric Depiereux (*Promoteur, FUNDP*)
Dr. Christophe Lambert (*BioXPR, Namur*)
Prof. Jean-Yves Matroule (Président du jury, FUNDP)
Prof. Michel Milinkovitch (Co-p*romoteur, UNIGE, Suisse*)
Prof. Karine Van Doninck (*FUNDP*)
Prof. Jacques Van Helden (*ULB, Bruxelles*)

2010

TABLE DES MATIÈRES

INTRODUCTION GÉNÉRALE 6

PHYLOGÉNIE 6
INFÉRENCE PHYLOGÉNÉTIQUE 9
PHYLOGÉNIE ET GÉNOMIQUE COMPARATIVE 10

CADRE INFORMATIQUE POUR L'ESTIMATION DE PHYLOGÉNIES 14

INTRODUCTION 14
Méthodes d'inférence phylogénétique 14
Estimation d'une phylogénie par maximum de vraisemblance 18
Modèles de substitutions nucléotidiques 20
Calcul de la vraisemblance 23
Trouver une phylogénie optimale 25
Logiciels d'inférence phylogénétiques existants 26
Anciennes et nouvelles méta-heuristiques 29
Hill Climbing 30
Simulated Annealing 31
Algorithme génétique 32
MetaGA 33
MÉTHODOLOGIE ET CONCEPTION DU LOGICIEL METAPIGA 2.0 34
Langage de programmation utilisé 34
Fonctionnalités implémentées 36
Fonctionnalités de manipulation du jeu de données 37
Heuristiques 38
Simulated Annealing 38
Algorithme génétique 41
MetaGA 42
Critères d'évaluation 45
Génération d'un arbre de départ 47
Opérateurs de mutation 48
NNI (Nearest Neighbor Interchange) 49
SPR (Subtree Pruning and Regrafting) 49
TBR (Tree Bisection Reconnection) 51
TXS (TaXa Swap) 52
STS (SubTree Swap) 52
BLM et BLMint (Branch Length Mutation) 53
RPM (Rate Parameters Mutation) 53
GDM (Gamma Distribution Mutation) 53
PIM (Proportion of Invariant Mutation) 53
APRM (Among-Partition Rate Mutation) 54
Réplicats et conditions d'arrêt 54
Conditions d'arrêt des méta-heuristiques 56
Parallélisation 57
Outils d'analyse 58
Outils périphériques intégrés 59
Optimisation du calcul de la vraisemblance 61
METAPIGA 2.0 : MAXIMUM LIKELIHOOD LARGE PHYLOGENY ESTIMATION USING THE METAPOPULATION GENETIC ALGORITHM AND OTHER STOCHASTIC HEURISTICS 62
Title and abstract 62

CONCLUSIONS 64

CADRE PHYLOGÉNÉTIQUE POUR LA COMPARAISON DE GÉNOMES MULTI-ESPÈCES 66

INTRODUCTION 66
Événements de duplication et détermination des relations d'homologie 67
DÉVELOPPEMENT DE MANTIS 68
MANTiS: a phylogenetic framework for multi-species genome comparisons 70
Title and abstract 71
Méthodologie et conception du logiciel Mantis 72
Base de données 72
Requêtes 76
Statistiques de représentation d'une catégorie 79
Statistiques générales 80
UTILISATION DE MANTIS POUR ANALYSER L'ÉVOLUTION DES GÉNOMES 84
Mapping gene gains and losses among metazoan full genomes using an integrated phylogenetic framework 85
Title and abstract 86
2x genomes—depth does matter 87
Title and abstract 88
Historical constraints on vertebrate genome evolution 89
Title and abstract 90
CONCLUSIONS 92

CONCLUSIONS ET PERSPECTIVES 94

METAPIGA 2.0 94
MANTIS 95

RÉFÉRENCES 98

GÉNÉRALES 98
METAPIGA 2.0 : MAXIMUM LIKELIHOOD LARGE PHYLOGENY ESTIMATION USING THE METAPOPULATION GENETIC ALGORITHM AND OTHER STOCHASTIC HEURISTICS 103
MANTIS: A PHYLOGENETIC FRAMEWORK FOR MULTI-SPECIES GENOME COMPARISONS 106
MAPPING GENE GAINS AND LOSSES AMONG METAZOAN FULL GENOMES USING AN INTEGRATED PHYLOGENETIC FRAMEWORK 108
2X GENOMES—DEPTH DOES MATTER 111
HISTORICAL CONSTRAINTS ON VERTEBRATE GENOME EVOLUTION 113

INTRODUCTION GÉNÉRALE

PHYLOGÉNIE

Lorsque Charles Darwin publia sa théorie sur l'évolution des espèces, il fut le premier à suggérer une théorie convaincante (la sélection naturelle) expliquant le processus d'évolution, et confirmant que les êtres vivants descendent tous les uns des autres. Jusqu'aux années 1960, les biologistes classaient les espèces en comparant la morphologie, le comportement et la répartition géographique des espèces. Cette méthode traditionnelle de classification, fondée sur la notion de caractères[1] homologues[2], s'appelle la cladistique. Lorsque l'on découvrit que les variations des séquences d'acides aminés étaient conservées d'une espèce à l'autre et pouvaient être utilisées pour retracer des relations entre ces espèces, une nouvelle discipline était née : la phylogénie, qui est l'étude de la formation et de l'évolution des organismes vivants en vue d'établir leur parenté. La phylogénie moléculaire a ainsi permis de redonner un nouveau souffle à la science taxonomique en permettant de mieux comprendre les relations entre espèces. Le développement des techniques de séquençage des génomes a permis de considérer les séquences génétiques comme des caractères à part entière, et donc d'appliquer la cladistique aux séquences de nucléotides. L'approche bioinformatique utilisant ces données moléculaires a dans un premier temps confirmé les résultats d'analyses cladistiques traditionnelles, avant de réorganiser certaines phylogénies de façon plus surprenante. Ces réorganisations de l'arbre du vivant posent des questions incroyablement profondes du point de vue de l'évolution, car si la cladistique traditionnelle basée sur les caractères phénotypiques s'est trompée, cela signifie que certains caractères ou mécanismes de développement apparemment non homologues sont beaucoup plus proches qu'on ne le croit, ou inversement que des caractères apparemment homologues sont beaucoup plus éloignés qu'on ne le croit.

La phylogénie nous permet donc de nous pencher sur l'évolution de caractères génétiques pour aider à comprendre des mécanismes biologiques complexes, et peut également être associées à d'autres domaines de la biologie, certains ayant des applications très pratiques telles que la médecine légale ou les tests génétiques. Suite aux découvertes validées en phylogénie moléculaire, la systématique se concentre maintenant sur une classification phylogénétique (se basant donc sur les relations de parenté) plutôt que sur la classification qui n'utilisait que des critères de ressemblance phénotypiques. En effet, des espèces peuvent avoir une ressemblance phénotypique tout en étant éloignées génétiquement, en raison du phénomène de convergence évolutive[3]. La Figure 1 illustre ce phénomène en présentant

[1] **Caractère** : attribut observable d'un organisme.
[2] Des caractères sont dit **homologues** s'ils ont comme précurseur un caractère ancestral commun (voir Figure 4 page 14).
[3] **Convergence évolutive** : également appelée « homoplasie », il s'agit de la présence chez deux espèces de caractères analogues, d'une même adaptation, mais qui n'a pas été hérité d'un ancêtre commun. Elle résulte de deux évolutions indépendantes dans un même type

plusieurs espèces animales ayant de grandes ressemblances morphologiques, qui furent donc classées comme étant très proches avec la cladistique classique. La phylogénie moléculaire a permis de prouver que ces espèces faisaient en réalité partie de 2 groupes monophylétiques[4] distincts : les Afrothériens et les Laurasiathériens (Springer *et al.* 2004).

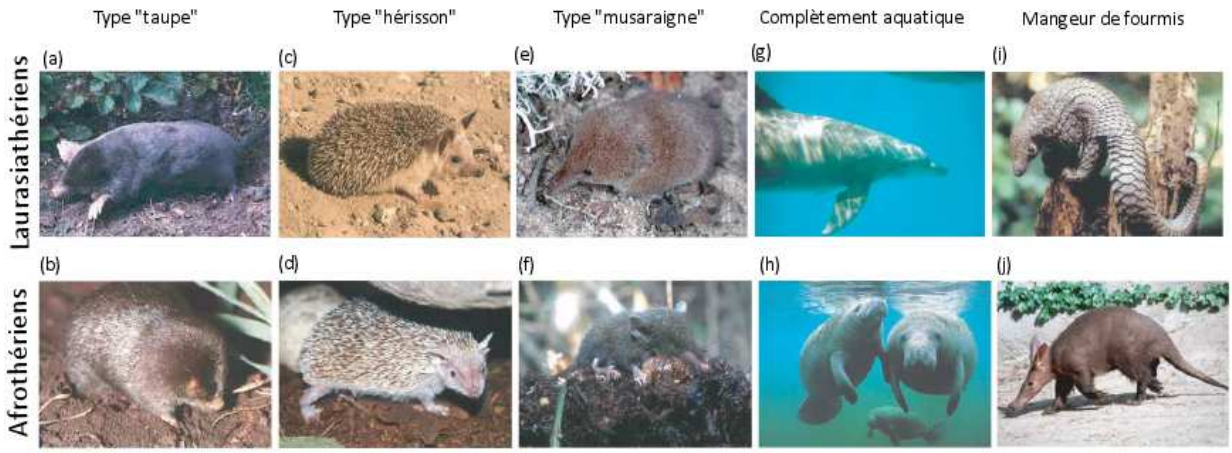

Figure 1 – Radiations morphologiques parallèles chez les Afrothériens et les Laurasiathériens, illustrant la convergence évolutive de la morphologie externe. (a) taupe d'Europe (Talpinae) ; (b) taupe dorée africaine (Chrysochlorinae) ; (c) hérisson commun (Erinaceinae) ; (d) hérisson malgache (Tenrecinae) ; (e) musaraigne commune (Soricinae) ; (f) tenrec musaraigne (Oryzorictinae) ; (g) dauphin (Delphininae) ; (h) lamantin (Trichechidae) ; (i) pangolin (Maninae) ; (j) oryctérope (Orycteropodidae). (figure reproduite de Springer *et al.* 2004).

Une phylogénie est représentée sous la forme d'un arbre phylogénétique, dont les nœuds représentent les taxa[5] (les feuilles de l'arbre représentant les espèces actuelles et les nœuds internes leurs ancêtres), la topologie représente les relations de parenté entre les taxa, et les longueurs de branches représentent la distance évolutive entre eux. Si cet arbre est raciné, une direction est donnée aux changements entre taxa, la racine représentant l'ancêtre commun le plus proche entre toutes les taxa se trouvant aux feuilles. En exemple, les 2 figures ci-dessous sont des arbres phylogénétiques représentant respectivement une phylogénie (racinée) probable de tous les êtres vivants (Figure 2) et la phylogénie des principales espèces actuelles de primates (Figure 3).

d'environnement (opposé à la synapomorphie qui désigne une similarité due à un ancêtre commun).

[4] **Monophylétique** : un groupe incluant un ancêtre commun et la totalité de ses descendants.

[5] **Taxon** : Groupe d'êtres vivants ou fossiles qui ont des traits communs. Il ne faut pas confondre le taxon, tel que défini ici, et les diverses catégories de la taxonomie, qui sont en fait des niveaux de classification (comme la famille, le genre, l'espèce, etc.). Les principales catégories de taxons sont les suivantes, de la plus grande à la plus petite : règne, embranchement (ou *phylum*), classe, ordre, famille, genre et espèce.

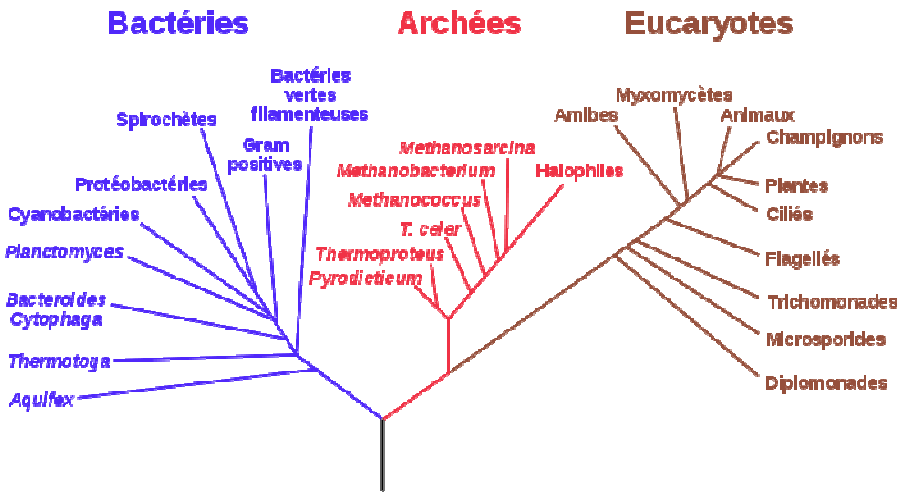

Figure 2 - Arbre phylogénétique hypothétique de tous les organismes vivants. L'arbre est basé sur des séquences de l'ARNr 16S. À l'origine proposé par Carl Woese, il montre l'histoire évolutive des trois domaines du vivant (bactéries, archaea et eucaryotes). (Woese 1998).

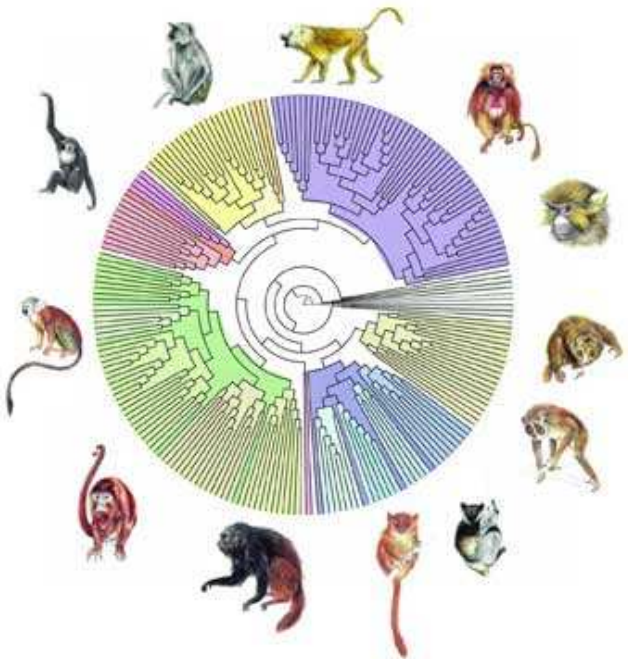

Figure 3 - Arbre phylogénétique incorporant 217 des 233 espèces actuelles de primates, reconstruit par l'algorithme PhyML (Guindon et Gascuel, 2003).

INFÉRENCE PHYLOGÉNÉTIQUE

L'inférence phylogénétique permet, à partir d'un ensemble de séquences alignées (qu'il s'agisse de nucléotides, de protéines ou même de séquences de caractères présents/absents), de générer un arbre phylogénétique qui puisse expliquer les relations de parenté existant dans ces données. Quelle que soit la méthode utilisée, l'inférence phylogénétique se base toujours sur un modèle de l'évolution des séquences, et la qualité de l'arbre obtenu dépendra bien sûr de l'adéquation du modèle utilisé. Il faut en effet tenir compte du fait qu'au cours de l'évolution, des mutations de l'ADN peuvent se produire, entraînant la transformation d'une base en une autre (on parle de substitution nucléotidique). En observant 2 séquences alignées d'espèces actuelles proches, il est donc difficile de déterminer combien de substitutions ont eu lieu depuis la séquence de leur ancêtre commun, nécessitant l'utilisation d'un modèle permettant de simuler ces événements.

Depuis les débuts de l'inférence phylogénétique, de nombreuses méthodes ont été développées (voir « Méthodes d'inférence phylogénétique » page 14), et si nombre d'entre elles ont plus ou moins été abandonnées aujourd'hui, il existe à l'heure actuelle une pléthore d'algorithmes et de logiciels permettant d'inférer un arbre phylogénétique à partir d'un jeu de données biologiques. Cependant, il n'existe aucun logiciel qui soit vraiment complet, regroupant en une seule suite d'outils informatiques les principales méthodes d'inférence actuelles. De plus, la plupart d'entre eux sont relativement complexes à utiliser, se basant sur des lignes de commandes codées (voir « Logiciels d'inférence phylogénétiques existants » page 26). Si c'était la norme il y a une vingtaine d'années, les utilisateurs s'attendent aujourd'hui au minimum à disposer d'une interface graphique pour interagir avec un logiciel.

Comme nous l'expliquerons dans la section suivante, nous avons choisi de nous concentrer sur les méthodes basées sur le maximum de vraisemblance[6]. Nous verrons que ce critère est le plus utilisé à l'heure actuelle, car il s'agit d'une méthode statistique robuste[7] et consistante[8], pouvant tenir compte de la complexité du processus de substitutions nucléotidiques. Actuellement, les logiciels les plus populaires basés sur le maximum de vraisemblance utilisent surtout des méthodes Bayesiennes[9] pour inférer des phylogénies. Nous avons donc choisi d'aborder le problème avec une autre approche populaire en informatique, les méta-heuristiques. Il s'agit d'une famille d'algorithmes pouvant résoudre des problèmes NP-Complexes, adaptables dans des domaines nombreux et variés, et qui pourraient offrir d'excellents résultats en inférence phylogénétique. Un problème NP-Complexe (tel que l'inférence phylogénétique) est un problème pour lequel trouver la meilleure solution nécessite un parcours exhaustif de toutes les solutions possibles, et dont le calcul de toutes ces

[6] **Maximum de vraisemblance** : méthode statistique qui vise à définir le meilleur estimateur pour un problème donné.
[7] **Robustesse** : capacité d'un estimateur statistique à ne pas être modifié par une petite modification dans les données ou dans les paramètres du modèle choisi pour l'estimation.
[8] **Consistance** : en phylogénétique, une méthode est consistante si elle converge vers le résultat correct lorsqu'elle dispose de suffisamment de données.
[9] **Méthodes Bayesiennes** : qualifie des méthodes d'inférences statistiques fondées sur une évaluation des probabilités des hypothèses, préalablement à l'observation d'un événement aléatoire.

solutions prendrait un temps infini, même si nous pouvions améliorer les capacités de calcul à notre disposition de manière exponentielle. Bien entendu, ces méta-heuristiques doivent être adaptées au problème à résoudre, et jusqu'ici peu d'entre elles l'ont été dans le domaine de l'inférence phylogénétique. À ce jour, quelques logiciels ont adapté un algorithme génétique (GARLI, *Zwickl 2006 ;* GAML, Lewis 1998 *;* Metapiga_v1, *Lemmon & Milinkovitch 2002*), un logiciel utilise l'algorithme du quartet puzzling (TreePuzzle, *Schmidt et al. 2002*), et il existe également une adaptation de Simulated Annealing (*Salter & Pearl 2001*). Notre but sera donc de développer un cadre informatique d'inférence phylogénétique, permettant d'adapter des méta-heuristiques ayant prouvé leurs performances dans d'autres domaines, et offrant aux biologistes une alternative intéressante aux méthodes d'inférence Bayesiennes. Nous accorderons également une importance particulière à la taille des jeux de données qu'il sera capable de traiter. En effet, nous avons choisi d'utiliser le maximum de vraisemblance qui est une méthode consistante : plus on ajoute de données, plus la vraisemblance converge vers la bonne topologie. L'abondance des données permettra donc d'obtenir de meilleures phylogénies. Parallèlement, les techniques de séquençages ayant beaucoup évoluées, les biologistes disposent également de plus en plus de données brutes, et seuls quelques logiciels (tel que MrBayes, *Huelsenbeck & Ronquist 2001 & 2003*) permettent à l'heure actuelle d'inférer efficacement une phylogénie basée sur un jeu donnée composé de centaines de séquences longues de plusieurs milliers de nucléotides.

Dans la première partie de cette thèse, nous plongerons dans l'inférence phylogénétique, présentant un petit état de l'art du domaine puis en développant un logiciel qui réponde aux critères que nous venons de présenter : (1) un outil informatique moderne et convivial, (2) qui regroupe les principales méthodes existantes à l'heure actuelle, (3) qui permette de travailler avec de très grands jeux de données et (4) qui offre un cadre informatique robuste permettant de développer et tester de nouvelles méta-heuristiques, qui auront été préalablement adaptées au problème de l'inférence phylogénétique.

PHYLOGÉNIE ET GÉNOMIQUE COMPARATIVE

La génomique comparative est l'étude comparative de la structure et fonction des génomes de différentes espèces, dont les buts principaux sont de mieux comprendre comment les différentes espèces ont évolué, quels sont les effets de la sélection sur l'organisation et l'évolution des génomes, ainsi que de déterminer les fonctions des gènes et des régions non-codantes du génome. Beaucoup de méthodes et de bases de données disponibles utilisent des approches de comparaison de génomes deux à deux, permettant de déterminer les relations d'homologies (voir Figure 4) deux à deux, ainsi que dans une certaine mesure les relations d'orthologie et de paralogie au sein des familles de gènes (mais limitées à cause du manque d'information phylogénétique) et les annotations fonctionnelles pouvant être translatées entre les deux génomes (par exemple si la fonction d'un gène est connue pour une espèce, elle peut être similaire pour l'orthologue de ce gène s'il existe pour une autre espèce). La génomique comparative peut cependant énormément bénéficier de la phylogénie. La comparaison directe de génomes fonctionne mal sur des paires de taxa très divergents, mais effectuer ces comparaisons au sein d'un cadre phylogénétique procure des références

intermédiaires (Dubchak & Frazer 2003). Ces références intermédiaires fournissent de l'information supplémentaire, qui peut être utile lors de l'annotation de génome par exemple. Les possibilités d'analyses offertes s'élargissent alors, permettant d'étudier l'histoire des duplications et des pertes de gènes, des réarrangements et recombinaisons de types variés, des pertes et des gains d'introns, des transferts de gènes horizontaux et la possibilité de reconstruire les génomes ancestraux présumés. Le coût computationnel de l'inférence phylogénétique assez élevé, associé aux difficultés de son interprétation, impliquent malheureusement que beaucoup de méthodes et bases de données sont toujours développées par comparaisons deux à deux, bien qu'il ait été prouvé que l'identification des orthologues et des paralogues basés sur la phylogénie soit l'approche la plus valide (Vilella *et al.* 2009 ; Alexeyenko *et al.* 2006 ; Li *et al.* 2003 ; Gabaldon 2008).

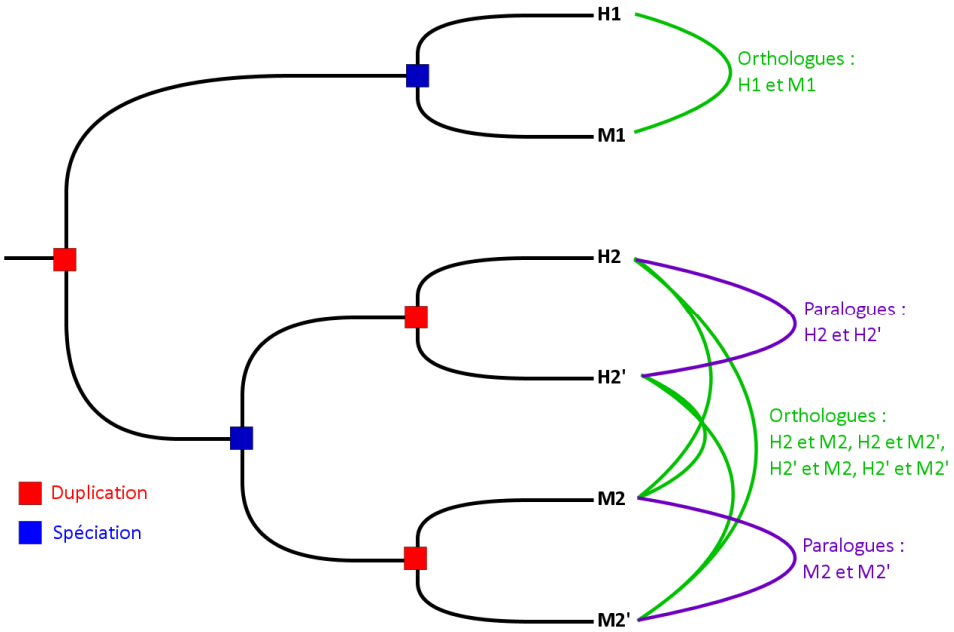

Figure 4 – Illustration des relations d'homologie dans un arbre de gènes. Des caractères sont dit homologues s'ils ont comme précurseur un caractère ancestral commun. Il existe 2 types d'homologies : on parle de paralogie lorsqu'une relation d'homologie entre une paire de gènes est générée par un événement de duplication, et on parle d'orthologie lorsqu'une relation d'homologie entre une paire de gènes est générée par un événement de spéciation. Cette figure montre l'évolution d'un hypothétique caractère après plusieurs événements de duplication et de spéciation. Les gènes actuels sont représentés sur les branches terminales, où H1, H2 et H2' sont des gènes humains et M1, M2, M2' sont des gènes appartenant à la souris.

Dans la seconde partie de cette thèse, nous mettons en avant une approche phylogénétique pour la génomique comparative, en développant un logiciel qui permettra d'explorer le contenu d'une sélection de génomes eucaryotes le long d'une phylogénie. Les relations d'orthologie et de paralogie seront identifiées à l'aide d'arbres phylogénétiques, et nous pourrons suivre l'historique des événements de duplication au cours de l'évolution et déterminer à quel moment un « caractère » a été gagné et perdu. Cette cartographie des gènes sur les branches de la phylogénie nous permettra également de reconstituer le contenu des génomes ancestraux présumés. Enfin, en associant à ces données différentes annotations fonctionnelles (telles que processus biologiques et fonctions moléculaires dans lesquels les

gènes sont impliqués, ou les tissus dans lesquels ils sont exprimés), nous pourrons étudier l'évolution fonctionnelle d'un génome au cours du temps. En se focalisant sur une branche de la phylogénie, et en comptant les gènes associés à une fonction donnée, nous pourrons déterminer si cette fonction est plus ou moins représentée par rapport au génome actuel. Enfin, comme pour le logiciel d'inférence phylogénétique, nous mettrons tout en œuvre pour que ce logiciel soit convivial et offre différents moyens d'exploration graphique des données. Nous tenterons également d'intégrer une interface permettant d'interagir dynamiquement avec l'ensemble des données, proposant à l'utilisateur de poser le plus simplement possible un vaste éventail de questions de génomique comparative avec des critères phylogénétiques, augmentant considérablement l'intérêt du logiciel pour la communauté scientifique.

CADRE INFORMATIQUE POUR L'ESTIMATION DE PHYLOGÉNIES

INTRODUCTION

Comme nous l'avons présenté dans l'introduction générale, la première partie de cette thèse abordera l'inférence phylogénétique, et nous discuterons comment l'informatique peut fournir des outils précieux à cette tâche complexe. Nous avons déjà introduit brièvement ce qu'est l'inférence phylogénétique et nous développerons ci-dessous comment inférer une phylogénie par maximum de vraisemblance, les difficultés que cela soulève et pourquoi utiliser une heuristique devient dès lors indispensable. Nous présenterons ensuite les logiciels les plus populaires permettant d'inférer des phylogénies par maximum de vraisemblance, ainsi que leurs limitations actuelles. Nous terminerons cette introduction en exposant l'intérêt d'implémenter de nouvelles heuristiques pour la recherche d'un arbre phylogénétique optimal, en détaillant les solutions que nous avons choisies.

MÉTHODES D'INFÉRENCE PHYLOGÉNÉTIQUE

L'inférence d'une phylogénie est une procédure permettant d'estimer le mieux possible une histoire évolutive basée sur des informations incomplètes, étant donné que nous ne disposons que de données moléculaires sur des organismes actuels (nous n'envisagerons pas ici les sources de données morphologiques, physiologiques, et comportementales). Plusieurs méthodes existent, qui peuvent être divisées en deux grandes familles : (1) les méthodes itératives permettant de générer un arbre en un nombre fini d'étapes, et (2) les méthodes basées sur un critère d'optimalité qui permet de comparer la « qualité » de solutions (phylogénies) différentes. Toutes ces méthodes se basent sur un alignement de séquences de caractères biologiques, généralement des nucléotides (nous nous limiterons à ce cas dans cette section), mais il peut également s'agir de protéines ou de la présence/absence de caractères. Nous allons rapidement survoler quelques-unes de ces méthodes.

Les algorithmes de « UPGMA[10] » (Michener & Sokal 1957) et « NJ[11] » (Saitou & Nei 1987) sont de bons exemples de la première famille de méthodes. Ils se basent sur une matrice de distances entre chaque taxon et la topologie initiale est dite « en étoile », c'est-à-dire un graphe où toutes les branches terminales sont rattachées à un unique nœud interne. Les deux taxa ayant la distance la plus courte entre eux définissent un nouveau nœud interne. Les distances entre ce nouveau nœud et tous les autres sont recalculées (de manière différente pour UPGMA et NJ). La nouvelle matrice de distances est utilisée pour définir un nouveau

[10] **UPGMA** : Unweighted Pair Group Method with Arithmetic Mean.
[11] **NJ** : Neighbor Joining.

nœud interne, et ainsi de suite jusqu'à ce que l'arbre soit strictement dichotomique. Au terme de l'algorithme, nous obtiendrons donc un arbre, le « UPGMA Tree » ou le « Neighbor Joining Tree » (NJT), c'est-à-dire une estimation de notre phylogénie basée sur le nombre de différences entre toutes les paires de séquences alignées. (voir Figure 5). Malheureusement, utiliser le nombre de différences observées entre deux séquences alignées séparées par un temps *t* est un indicateur très faible du nombre de substitutions qui se sont produites entre ces 2 séquences (à moins que *t* soit petit). La divergence entre deux séquences n'augmente pas linéairement avec le temps car plusieurs substitutions peuvent se produire sur le même site[12], même pour des séquences de longueur infinie. Il est même possible que la similarité entre deux séquences puisse augmenter localement dans le temps. Il est également possible qu'un site observé aujourd'hui ait le même état que dans la séquence ancestrale, mais ait subit un certain nombre de substitutions avant de revenir à l'état initial. Des modèles de changements évolutifs ont donc été proposés (Jukes & Cantor 1969 ; Kimura 1980 ; Hasegawa *et al.* 1985 ; Tavaré 1986 ; Tamura & Nei 1993) pour approximer la dynamique des substitutions nucléotidiques, permettant de déterminer la probabilité qu'un caractère soit toujours dans le même état après un temps *t*. En utilisant l'un de ces modèles pour corriger la distance entre deux séquences, nous pourrons générer une nouvelle matrice de distances, et obtenir une meilleure estimation de la phylogénie avec les méthodes algorithmiques du type « UPGMA » ou « NJ ». Nous ne discuterons pas ici en détail de la nette supériorité de l'algorithme NJ par rapport à celui de l'UPGMA. Très succinctement, il suffit de savoir que l'algorithme UPGMA part de l'hypothèse (hautement irréaliste) que les taux de substitutions sont constants dans toutes les branches de l'arbre, c'est-à-dire postuler l'existence d'une horloge moléculaire[13] ponctuelle (Kimura 1968).

[12] **Site** : une position relative (« colonne ») dans un alignement de séquences.

[13] **Horloge moléculaire** : hypothèse selon laquelle les protéines homologues évoluent à la même vitesse quelles que soient les contraintes exercées par l'environnement.

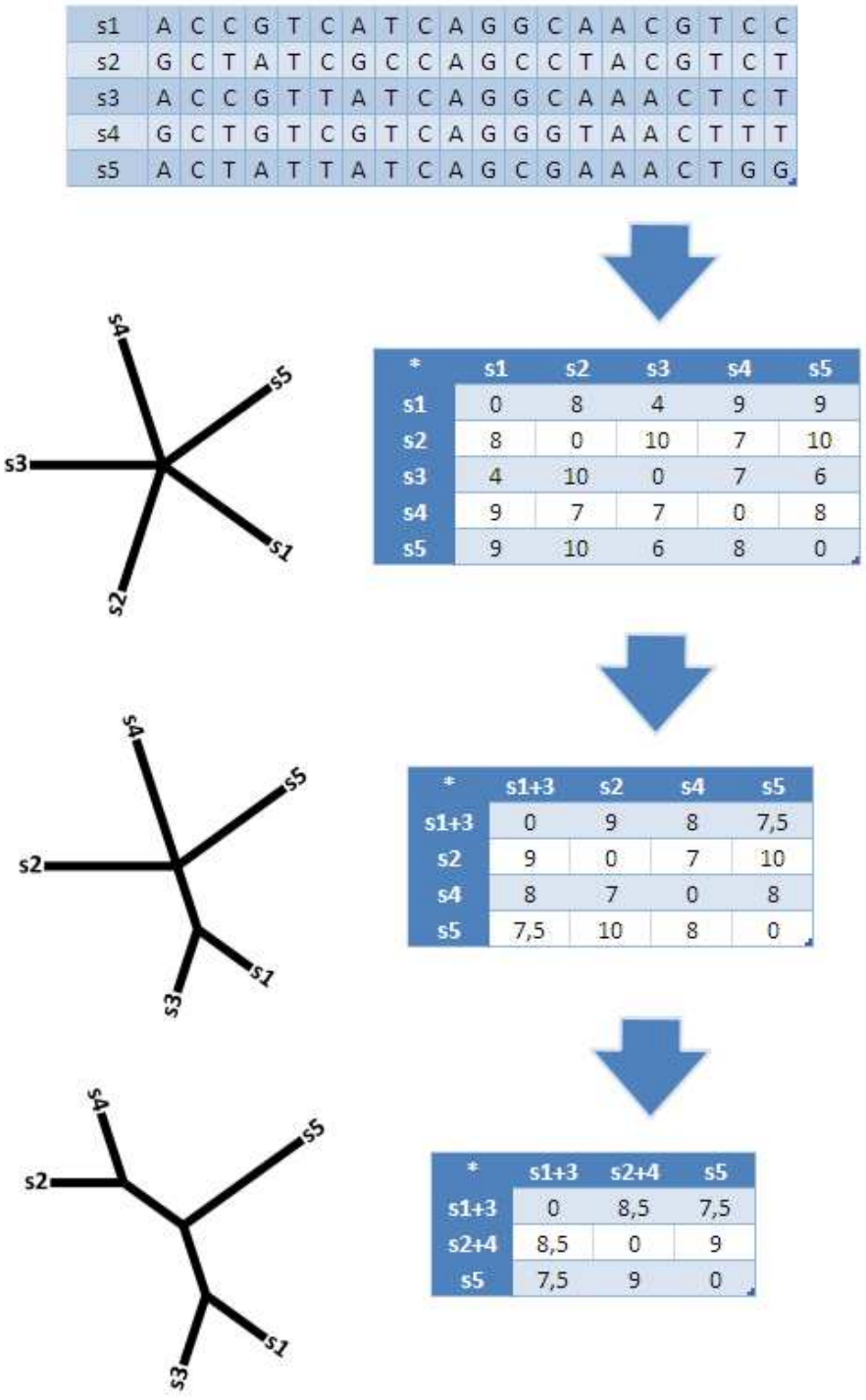

Figure 5 – Algorithme de l'UPGMA. En nous basant sur l'alignement des 5 séquences de nucléotides, nous construisons une matrice de distance (en comptant le nombre de différences entre chaque paire de séquences) et créons une topologie initiale en étoile. La distance minimale de la première matrice est 4 (entre s1 et s3). Un nouveau nœud interne est alors créé dans la topologie, regroupant s1 et s3, et une nouvelle matrice de distance est générée (dans laquelle nous avons regroupé les lignes s1 et s3 en une ligne s1+3, et pour laquelle nous avons recalculé les distances). En répétant cette procédure une seconde fois, nous obtenons la topologie finale de l'arbre UPGMA.

Les algorithmes itératifs tels que le NJ sont très rapides et pour un modèle choisi ils produiront toujours la même phylogénie à partir d'un ensemble de séquences donné. Cependant, aucun critère objectif ne permet d'évaluer la différence de qualité entre deux arbres.

Ceci nous amène à la deuxième famille de méthodes qui utilisent un critère d'optimalité pour comparer différents arbres entre eux afin de déterminer le meilleur (topologie + longueurs des branches) (Felsenstein 2004). Formellement, ce critère d'optimalité est décrit comme une fonction objectif[14], qui nous fournira pour chaque arbre un score. Il nous suffira ensuite d'appliquer un algorithme permettant de générer des topologies afin d'optimiser cette fonction.

Trois grands critères d'optimalité sont classiquement employés en phylogénie : l'évolution minimum, la parcimonie, et le maximum de vraisemblance. Nous ignorerons ici le critère d'évolution minimum (basé sur le principe des moindres carrés, Rzhetsky and Nei 1993) car il a le désavantage d'utiliser des matrices de distances (il y a perte d'information lorsque l'on construit une matrice de distance puisqu'il est impossible de reconstruire l'alignement initial). Les deux autres critères fonctionnent directement sur les données brutes (l'alignement multiple de séquences). Le principe du maximum de parcimonie est de privilégier le scénario impliquant le minimum de changements évolutifs : le meilleur arbre (topologie uniquement) est celui qui nécessite le minimum de mutations pour expliquer les séquences observées (Camin & Sokal 1965) ; il s'agit donc de minimiser la « longueur totale de l'arbre ». Cette méthode fut parmi les premières et les plus utilisées par les biologistes pour estimer une phylogénie (Felsenstein 2004). Elle a cependant de grandes faiblesses, et est de moins en moins utilisée actuellement, au profit des méthodes basées sur des modèles de changements évolutifs. En effet, la méthode de maximum de parcimonie n'est efficace que lorsque le nombre de substitutions est faible, c'est-à-dire lorsque les substitutions multiples sont négligeables. Par contre, le principe de maximum de vraisemblance nécessite d'évaluer la probabilité qu'un arbre (topologie ET longueurs de branches) ait généré les données observées. Cette méthode nécessite donc d'utiliser un modèle de substitution (permettant de tenir compte des substitutions multiples). Nous nous trouvons donc dans un cadre statistique consistant. En plus de ses propriétés de consistance, le maximum de vraisemblance est généralement moins affecté par des erreurs d'échantillonnage, et résiste mieux aux éventuelles violations des hypothèses prises dans les modèles (Schadt *et al.* 1998). C'est donc un excellent estimateur, que nous avons choisi dans notre projet d'estimation de phylogénies.

[14] **Fonction objectif** : le terme fonction objectif est utilisé en optimisation (mathématiques, informatique) pour désigner une fonction qui sert de critère pour déterminer la meilleure solution à un problème d'optimisation. Concrètement, une fonction objectif associe une valeur à une instance d'un problème d'optimisation. Le but du problème d'optimisation est alors de minimiser/maximiser cette fonction.

ESTIMATION D'UNE PHYLOGÉNIE PAR MAXIMUM DE VRAISEMBLANCE

Le maximum de vraisemblance étant à présent choisi comme méthode d'estimation, voyons en détails comment calculer la vraisemblance d'un arbre selon les principaux modèles de substitutions nucléotidiques.

Tous les modèles que nous allons utiliser sont des modèles Markoviens : ils impliquent que la probabilité d'un changement d'un état i à un état j pour un site particulier ne dépend pas de l'histoire de ce site avant l'état i. Ils font également l'hypothèse que les probabilités de substitutions instantanées sont les mêmes pour chaque partie de l'arbre, constituant ainsi un processus de Markov homogène[15].

Ces modèles sont réversibles dans le temps (« time-reversible »), assumant que le taux global de changement d'un état i à un état j dans une période de temps donnée est le même que le taux de changement de l'état j à l'état i. Grâce à cette hypothèse, peut importe le nœud de l'arbre que nous choisissons comme racine, la vraisemblance sera toujours la même. Nous pouvons dès lors sélectionner arbitrairement un nœud comme racine, et sous l'hypothèse que différents sites nucléotidiques évoluent indépendamment, calculer séparément la vraisemblance pour chacun d'eux à la racine, et combiner ces valeurs à la fin. Pour calculer la vraisemblance d'un site, nous devons considérer tous les scénarios d'évolution possibles des séquences au niveau des feuilles, et sommer la probabilité de chacun d'eux.

L'expression mathématique d'un modèle de substitution est une table des taux instantanés (substitutions par site par unité de distance évolutive) dans laquelle chaque nucléotide est remplacé par chaque autre nucléotide possible. De manière générale, nous utiliserons la matrice de taux instantanés $\boldsymbol{Q}$, dans laquelle chaque élément $\boldsymbol{Q}_{ij}$ représente le taux de changement de la base i à la base j pendant une période temps infinitésimale dt.

La forme la plus générale de cette matrice est :

$$\boldsymbol{Q} = \begin{bmatrix} \cdot & \mu a \pi_C & \mu b \pi_G & \mu c \pi_T \\ \mu g \pi_A & \cdot & \mu d \pi_G & \mu e \pi_T \\ \mu h \pi_A & \mu i \pi_C & \cdot & \mu f \pi_T \\ \mu j \pi_A & \mu k \pi_C & \mu l \pi_G & \cdot \end{bmatrix} \qquad \text{Eq.1}$$

Où π_X représente la fréquence à l'équilibre de la base X, et μ le taux instantané moyen de substitution. Ce taux moyen est modifié par les paramètres de taux relatifs $a, b, c, \dots, l$, qui correspondent à chaque transformation possible d'une base à une base différente. Étant donné que les modèles utilisés sont réversibles dans le temps, nous pouvons d'emblée considérer pour la suite que $g = a$, $h = b$, $i = c$, $j = d$, $k = e$, $l = f$. Les « $\cdot$ » dans la matrice sont égaux à la négation de somme des autres composantes de cette ligne. Le « $\cdot$ » en (i, j) est donc égal à $-\sum_{i \neq j} i$.

[15] **Processus de Markov** : Processus probabiliste selon lequel un système passe d'un état à un autre (parmi un nombre fini ou infini) à intervalles de temps réguliers. L'évolution du processus ne dépend que de l'état dans lequel il se trouve présentement. Tout le passé de l'évolution du processus se trouve donc résumé dans son état au dernier instant où on le connaît.

Nous pouvons calculer le taux instantané moyen de substitution de la manière suivante :

$$\mu = \frac{1}{\sum_{i \neq j}^{A,C,T,G} \pi_i \boldsymbol{Q'}_{ij}} \qquad \textbf{Eq.2}$$

Où

$$\boldsymbol{Q'} = \begin{bmatrix} - & a\pi_C & b\pi_G & c\pi_T \\ a\pi_A & - & d\pi_G & e\pi_T \\ b\pi_A & d\pi_C & - & f\pi_T \\ c\pi_A & e\pi_C & f\pi_G & - \end{bmatrix} \qquad \textbf{Eq.3}$$

$\boldsymbol{Q}$ pouvant être décomposée en 2 matrices, nous ferons parfois référence à $\boldsymbol{R}$, la matrice de taux, et à $\boldsymbol{\Pi}$, la matrice des fréquences à l'équilibre **:**

$$\boldsymbol{Q} = \boldsymbol{R} \times \boldsymbol{\Pi} \qquad \textbf{Eq.4}$$

$$\boldsymbol{R} = \begin{bmatrix} - & \mu a & \mu b & \mu c \\ \mu g & - & \mu d & \mu e \\ \mu h & \mu i & - & \mu f \\ \mu j & \mu k & \mu l & - \end{bmatrix} \qquad \textbf{Eq.5}$$

$$\boldsymbol{\Pi} = \begin{bmatrix} \pi_A & 0 & 0 & 0 \\ 0 & \pi_C & 0 & 0 \\ 0 & 0 & \pi_G & 0 \\ 0 & 0 & 0 & \pi_T \end{bmatrix} \qquad \textbf{Eq.6}$$

Pour calculer la vraisemblance d'un arbre, nous allons avoir besoin des probabilités de changement d'un état vers n'importe quel autre état le long d'une branche de longueur t. Cette matrice de transition, $\boldsymbol{P}(t)$, est calculée à partir de la matrice de taux instantané $\boldsymbol{Q}$ comme suit :

$$\boldsymbol{P}(t) = e^{\boldsymbol{Q}t} \qquad \textbf{Eq.7}$$

L'exponentielle peut être évaluée en décomposant $\boldsymbol{Q}$ en ses vecteurs propres et valeurs propres. Cependant, pour beaucoup de modèles, il existe des expressions simples des valeurs propres de $\boldsymbol{Q}$, permettant un calcul analytique direct des éléments de $\boldsymbol{P}(t)$.

Avant de parcourir plus en détail les modèles de substitution nucléotidique, il faut encore prendre en considération que chaque site n'évolue pas forcément à la même vitesse. Attention, ceci signifie que nous considérons que différents sites[16] peuvent évoluer à des vitesses différentes : par exemple, il est clair que les secondes positions de codons dans les séquences codant pour des protéines évoluent beaucoup plus lentement que les troisièmes positions de codons. Cette hétérogénéité des taux « accross characters » n'a donc rien à voir avec une hétérogénéité des taux à travers l'arbre, dont on tient déjà compte grâce à la possibilité de considérer dans le calcul de vraisemblance des branches de tailles variables. Il a été clairement montré qu'ignorer l'hétérogénéité des taux « accross characters » peut rendre l'inférence par maximum de vraisemblance inconsistante lorsque le vrai processus évolutif montrait des variations de taux site-à-site, et ce même si tous les autres aspects du processus sont modélisés correctement. Nous allons donc incorporer cette hétérogénéité dans le calcul

[16] **Site** : une position relative (« colonne ») dans un alignement de séquences.

de la vraisemblance en incluant un paramètre additionnel de taux relatif (r) dans l'expression de probabilité de transition. Dès lors, différentes vitesses d'évolution peuvent être assignées pour différents sous-ensembles de séquences, que nous appellerons catégories. Le taux relatif r est ajusté pour que le taux de substitution moyen soit égal à 1, et que donc les longueurs de branches reflètent toujours le nombre de substitutions moyen par site. La distribution la plus souvent utilisée pour modéliser l'hétérogénéité des taux est la distribution gamma. La distribution gamma a deux paramètres, un paramètre de forme α et un paramètre de taux β. En fixant $\beta = {}^{1}/_{\alpha}$, une distribution avec un taux moyen de 1 est obtenue, et une grande variété de distributions de taux peut être obtenue en variant simplement le paramètre α. Nous discrétisons la distribution gamma en la divisant en k catégories, avec une probabilité constante de $1/k$ de se trouver dans chaque catégorie. Nous pouvons dès lors calculer le point de pourcentage (le « cutting point ») de la distribution gamma (avec un paramètre donné α, et $\beta = {}^{1}/_{\alpha}$) pour la catégorie c comme suit :

$$z_{\Gamma}\left(\frac{c}{k}, \alpha\right) = \frac{z_{\chi^2}\left(\frac{c}{k}, 2\alpha\right)}{2\alpha} \qquad \text{Eq.8}$$

Où $z_{\chi^2}(p, v)$ est le point de pourcentage de la distribution χ^2 avec v degrés de liberté. Notez que $z_{\Gamma}(0, \alpha) = 0$.

Ensuite le taux de la catégorie c (c allant de 0 à $k-1$) est calculé de cette manière :

$$r(c) = \frac{I\left(\alpha+1, \alpha \times z_{\Gamma}\left(\frac{c+1}{k}, \alpha\right)\right) - I\left(\alpha+1, \alpha \times z_{\Gamma}\left(\frac{c}{k}, \alpha\right)\right)}{{}^{1}/_{k}} \qquad \text{Eq.9}$$

Où $I(z, \alpha) = \frac{1}{\Gamma(\alpha)} \int_0^z e^{-x} x^{\alpha-1} dx$ est la fonction gamma incomplète (DiDonato & Morris 1986).

MODÈLES DE SUBSTITUTIONS NUCLÉOTIDIQUES

Maintenant que nous avons vu comment calculer la matrice de transition $\boldsymbol{P}(t)$ depuis une matrice de taux instantanés $\boldsymbol{Q}$, tout en intégrant l'hétérogénéité des taux dans le calcul de la vraisemblance, nous pouvons passer en revue les différents modèles de substitutions nucléotides que nous utiliserons.

Le modèle le plus simple est celui qui a été proposé par Jukes et Cantor en 1969 (JC) (Jukes Cantor 1969). Il prend comme hypothèse que les fréquences à l'équilibre de toutes les bases sont égales et constantes ($\pi_A = \pi_C = \pi_G = \pi_T = 0.25$) et que les taux relatifs instantanés sont identiques ($a = b = c = d = e = f = 1$), le modèle n'ayant donc qu'un unique paramètre, généralement appelé α.

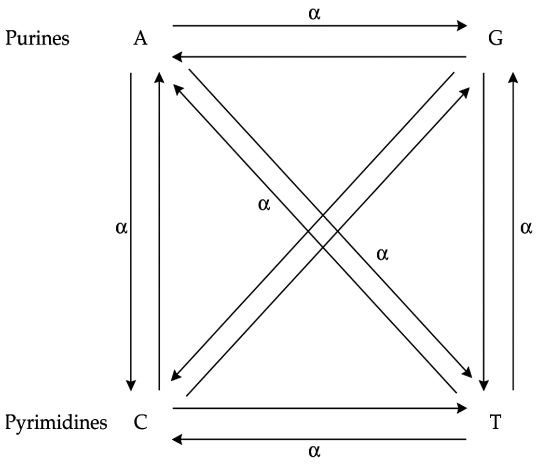

Figure 6 – Graphe d'états du modèle Jukes Cantor.

La matrice de taux instantanés a dès lors la forme suivante :

$$\boldsymbol{Q} = \begin{bmatrix} \cdot & \frac{1}{4}\mu & \frac{1}{4}\mu & \frac{1}{4}\mu \\ \frac{1}{4}\mu & \cdot & \frac{1}{4}\mu & \frac{1}{4}\mu \\ \frac{1}{4}\mu & \frac{1}{4}\mu & \cdot & \frac{1}{4}\mu \\ \frac{1}{4}\mu & \frac{1}{4}\mu & \frac{1}{4}\mu & \cdot \end{bmatrix} \qquad \textbf{Eq.10}$$

Avec un taux instantané moyen $\mu = {}^{4}/_{3}$. Pour ce modèle, une expression simple des valeurs propres de $\boldsymbol{Q}$ permet d'obtenir facilement les éléments de la matrice de transition :

$$P_{ij}(t) = \begin{cases} \frac{1}{4} + \frac{3}{4}e^{-\mu t r} & (i = j) \\ \frac{1}{4} - \frac{1}{4}e^{-\mu t r} & (i \neq j) \end{cases} \qquad \textbf{Eq.11}$$

Où r est le taux de la catégorie courante, en utilisant une hétérogénéité des taux.

Le modèle « Kimura 2 parameters » (K2P) (Kimura 1980), est un peu plus complexe, en prenant en considération le fait que les transitions [17] se produisent à un taux différent des transversions [18] . Il introduit le paramètre κ qui représente le ratio de taux transition:transversion, les paramètres de $\boldsymbol{Q}$ étant réduit à $a = c = d = f = 1$ et $b = e = \kappa$. Comme pour le modèle de Jukes et Cantor, il fait l'hypothèse que les fréquences à l'équilibre de toutes les bases sont égales et constantes ($\pi_A = \pi_C = \pi_G = \pi_T = 0.25$).

$$\boldsymbol{Q} = \begin{bmatrix} \cdot & \frac{1}{4}\mu & \frac{1}{4}\mu\kappa & \frac{1}{4}\mu \\ \frac{1}{4}\mu & \cdot & \frac{1}{4}\mu & \frac{1}{4}\mu\kappa \\ \frac{1}{4}\mu\kappa & \frac{1}{4}\mu & \cdot & \frac{1}{4}\mu \\ \frac{1}{4}\mu & \frac{1}{4}\mu\kappa & \frac{1}{4}\mu & \cdot \end{bmatrix} \qquad \textbf{Eq.12}$$

[17] **Transitions** : substitutions entre les états A et G, et entre les états C et T.
[18] **Transversions** : substitutions entre les états A et C, A et T, C et G, G et T.

Avec un taux instantané moyen $\mu = \frac{2+\kappa}{4}$. Pour ce modèle également, il existe une forme analytique permettant d'obtenir les éléments de la matrice de transition :

$$P_{ij}(t) = \begin{cases} \frac{1}{4} + \frac{1}{4}e^{-\mu tr} + \frac{1}{2}e^{-\mu tr\left(\frac{\kappa+1}{2}\right)} & (i = j) \\ \frac{1}{4} + \frac{1}{4}e^{-\mu tr} - \frac{1}{2}e^{-\mu tr\left(\frac{\kappa+1}{2}\right)} & (i \neq j, transition) \\ \frac{1}{4} - \frac{1}{4}e^{-\mu tr} & (i \neq j, transversion) \end{cases} \quad \text{Eq.13}$$

Si nous modifions le modèle de Kimura en faisant l'hypothèse que les fréquences à l'équilibre des bases peuvent être différentes, nous obtenons le modèle Hasegawa-Kishino-Yano proposé en 1985 (HKY85) (Hasegewa, Kishino & Yano, 1985). Nous fixerons dès lors les valeurs de $\boldsymbol{\Pi}$ aux fréquences des bases telles qu'elles sont observées dans les séquences de départ. La matrice de taux instantané devient donc :

$$\boldsymbol{Q} = \begin{bmatrix} \cdot & \mu\pi_C & \mu\kappa\pi_G & \mu\pi_T \\ \mu\pi_A & \cdot & \mu d\pi_G & \mu\kappa\pi_T \\ \mu\kappa\pi_A & \mu\pi_C & \cdot & \mu\pi_T \\ \mu\pi_A & \mu\kappa\pi_C & \mu\pi_G & \cdot \end{bmatrix} \quad \text{Eq.14}$$

Et le taux instantané moyen μ dépend des valeurs de $\boldsymbol{\Pi}$. Les éléments de la matrice de transitions sont calculés comme suit :

$$P_{ij}(t) = \begin{cases} \pi_j + \pi_j\left(\frac{1}{\Pi_j} - 1\right)e^{-\mu tr} + \left(\frac{\Pi_j - \pi_j}{\Pi_j}\right)e^{-\mu tr\left(1+\Pi_j(\kappa-1)\right)} & (i = j) \\ \pi_j + \pi_j\left(\frac{1}{\Pi_j} - 1\right)e^{-\mu tr} - \left(\frac{\pi_j}{\Pi_j}\right)e^{-\mu tr\left(1+\Pi_j(\kappa-1)\right)} & (i \neq j, transition) \\ \pi_j(1 - e^{-\mu tr}) & (i \neq j, transversion) \end{cases} \quad \text{Eq.15}$$

Où $\Pi_j = \pi_A + \pi_G$ si la base j est une purine (A ou G) et $\Pi_j = \pi_C + \pi_T$ si la base j est une pyrimidine (C ou T).

En 1993, Tamura et Nei ont complexifié ce modèle (TN93) (Tamura & Nei, 1993), en considérant que les transitions entre purines survenaient à un taux différent des transitions entre pyrimidines. Les paramètres de la matrice de taux instantané deviennent $a = c = d = f = 1$, $b = \kappa_1$ et $e = \kappa_2$:

$$\boldsymbol{Q} = \begin{bmatrix} \cdot & \mu\pi_C & \mu\kappa_1\pi_G & \mu\pi_T \\ \mu\pi_A & \cdot & \mu d\pi_G & \mu\kappa_2\pi_T \\ \mu\kappa_1\pi_A & \mu\pi_C & \cdot & \mu\pi_T \\ \mu\pi_A & \mu\kappa_2\pi_C & \mu\pi_G & \cdot \end{bmatrix} \quad \text{Eq.16}$$

Nous n'utiliserons pas de forme analytique pour calculer les valeurs de $\boldsymbol{P}(t)$, mais décomposerons $\boldsymbol{Q}$ en vecteurs et valeurs propres (voir modèle GTR ci-dessous).

Le dernier modèle que nous prendrons en considération est le plus général, le « General Time Reversible model » (GTR) (Tavaré, 1986). Il considère la forme la plus générale de $\boldsymbol{Q}$ réversible dans le temps :

$$\boldsymbol{Q} = \begin{bmatrix} \cdot & \mu a \pi_C & \mu b \pi_G & \mu c \pi_T \\ \mu a \pi_A & \cdot & \mu d \pi_G & \mu e \pi_T \\ \mu b \pi_A & \mu d \pi_C & \cdot & \mu f \pi_T \\ \mu c \pi_A & \mu e \pi_C & \mu f \pi_G & \cdot \end{bmatrix} \qquad \textbf{Eq.17}$$

Il n'y a pas de forme analytique pour calculer les probabilités de substitution, nous calculerons donc $\boldsymbol{P}(t)$ numériquement de cette façon :

$$\boldsymbol{P}(t) = \Omega \times \Phi(\mathrm{t}) \times \Omega^{-1} \qquad \textbf{Eq.18}$$

Où Ω est la matrice contenant les vecteurs propres de $\boldsymbol{Q}$, et avec :

$$\Phi_{\mathrm{ij}}(\mathrm{t}) = \begin{cases} e^{tr\Psi_{ii}} & (\mathrm{i} = \mathrm{j}) \\ 0 & (i \neq j) \end{cases} \qquad \textbf{Eq.19}$$

Où Ψ est la matrice diagonale contenant les valeurs propres de $\boldsymbol{Q}$.

CALCUL DE LA VRAISEMBLANCE

La méthode permettant de calculer la vraisemblance d'un arbre donné est une procédure récursive qui démarre d'une hypothétique racine sélectionnée à n'importe quel endroit de l'arbre, et combine la vraisemblance de chacun de ses sous-arbres descendants. Comme nous travaillons avec des modèles réversibles dans le temps, le choix de la racine ne changera pas la vraisemblance de l'arbre. Nous allons sommer la vraisemblance individuelle de chaque site, impliquant que la vraisemblance diminue lorsque le nombre de sites augmente, et amenant des nombres très petits, difficiles à manipuler. Pour parer à cela, nous utiliserons systématiquement le logarithme négatif de la vraisemblance. C'est à ce niveau-ci que nous allons introduire la possibilité que certains sites soient invariants (c'est-à-dire que ces sites ne puissent pas changer au cours du temps). Il s'agit d'une forme d'hétérogénéité des taux, mais qui plutôt qu'être modélisée par une distribution gamma, utilise uniquement 2 catégories distinctes : une catégorie où les sites varient normalement avec un même taux, et une seconde catégorie où les sites ne varient pas du tout (cela peut être dû, par exemple, à de fortes contraintes fonctionnelles). La proportion exacte de sites invariants peut être estimée et optimisée. Il est tout à fait possible d'utiliser un modèle d'hétérogénéité des taux en combinant invariant et distribution gamma (Gu. et al 1995 et Waddell & Penny 1996) : une partie des sites est invariable, tandis que les autres taux sont distribués selon une distribution gamma de paramètre α. Dans le calcul de la vraisemblance, nous noterons p_i la proportion de sites invariants.

Le calcul de la vraisemblance d'un arbre est alors effectué comme suit :

$$ML(arbre) = -\sum_{s=1}^{nSites} \ln\big((1-p_i)\Lambda_V(s) + p_i\Lambda_I(s)\big) \quad \text{Eq.20}$$

Où $nSites$ est le nombre de sites dans l'alignement de séquences, $\Lambda_I(s)$ est la vraisemblance invariable et $\Lambda_V(s)$ est la vraisemblance variable. La vraisemblance invariable est calculée comme suit :

$$\Lambda_I(s) = \begin{cases} \pi_i & si\ s\ est\ un\ site\ invariant\ ayant\ l'état\ i \\ 0 & si\ s\ n'est\ pas\ un\ site\ invariant \end{cases} \quad \text{Eq.21}$$

Et la vraisemblance variable est calculée comme suit :

$$\Lambda_V(s) = \sum_{i}^{A,C,G,T} \frac{\sum_{c=0}^{k} L_c(x_{Rs}=i)}{k}\pi_i \quad \text{Eq.22}$$

Où k est le nombre total de catégories utilisées pour l'hétérogénéité des taux, et R est le nœud sélectionné comme racine de l'arbre.

$L_c(x_{Rs}=i)$ est la vraisemblance conditionnelle de l'état i au site s pour la catégorie c à la racine, détaillée ci-dessous.

Si le nœud A est l'ancêtre qui a généré les séquences B et C, alors la vraisemblance conditionnelle de l'état i au site s pour la catégorie c dans A est

$$L_c(x_{As}=i) = \left[\sum_{j}^{A,C,T,G} P_{ij}\left(\frac{v_{AB}}{1-p_i}\right).L_c(x_{Bs}=j)\right] \times \left[\sum_{k}^{A,C,T,G} P_{ik}\left(\frac{v_{AC}}{1-p_i}\right).L_c(x_{Cs}=k)\right] \quad \text{Eq.23}$$

Où v_{xy} est la longueur de la branche joignant la séquence x à la séquence y.

TROUVER UNE PHYLOGÉNIE OPTIMALE

Disposant d'un estimateur et d'une évaluation de la vraisemblance d'un arbre sous différents modèles de substitutions nucléotidiques, nous pouvons aborder la recherche de la phylogénie optimale pour un alignement de séquences données. La vraisemblance nous informant sur la qualité d'une phylogénie (la probabilité que cette phylogénie ait généré les séquences observées), il nous « suffit » de trouver l'arbre qui maximise la vraisemblance. Si nous pouvions générer tous les arbres possibles pour un ensemble de données, et calculer la vraisemblance de chacun d'eux, il nous suffirait de sélectionner celui qui a la meilleure vraisemblance. Cette approche naïve n'est malheureusement pas applicable, car même en nous limitant aux différentes topologies, le nombre de possibilités croît factoriellement. En effet, le nombre de topologies (non-racinées[19]) différentes pour un ensemble de n séquences est égal à

$$T(n) = \frac{(2n-5)!}{2^{n-3}(n-3)!} \qquad \text{Eq.24}$$

La Figure 7 montre que dès 50 séquences, $T(n)$ dépasse les 10^{72} topologies différentes, et nous n'avons pris en compte ici *que* les différentes topologies. En effet, il faut également prendre en compte les possibilités des longueurs de chaque branche et de la valeur des paramètres du modèle (en utilisant GTR par exemple, il faut pouvoir estimer les 5 paramètres de taux, la forme de la distribution gamma si les taux sont hétérogènes, et la proportion de sites invariants). La complexité du calcul de la vraisemblance d'un arbre donné (avec sa topologie, ses longueurs de branches, et ses paramètres de modèles fixés) ne doit pas non plus être négligée ; il s'agit d'une procédure récursive[20], dont la complexité augmente non seulement avec le nombre de taxa, mais aussi avec la longueur des séquences. Aucun algorithme connu n'est capable de résoudre un problème aussi large en un temps polynomial : il s'agit d'un problème NP-complexe.

Il est cependant clair que le besoin de pouvoir évaluer des grandes phylogénies est bien présent. Premièrement, la précision de l'inférence augmente avec la taille de l'échantillonnage (le nombre de taxa et la longueur des séquences). Ensuite, le nombre de séquences ADN à disposition des chercheurs ne cesse d'augmenter, et les nouvelles questions qui se posent en biologie évolutive (c'est-à-dire les familles multi-géniques) nécessitent de travailler avec des grands ensembles de données.

[19] **Arbre non-raciné** : Dans un arbre non-raciné, le nombre de nœuds internes est égal au nombre de feuilles moins 2, et chaque nœud interne a toujours 3 nœuds voisins. Nous pouvons ensuite raciner un tel arbre sans en modifier la topologie en spécifiant que l'un des nœuds internes est l'ancêtre commun à tous les autres (ce nœud devient la racine de l'arbre). Il arrive également que des arbres phylogénétiques soient représentés par des arbres binaires, dans lesquels le nœud racine n'a que 2 voisins et aucun ancêtre.

[20] **Récursif :** En informatique, qualifie une procédure ou un programme qui s'utilise elle-même dans un traitement.

Heureusement, les informaticiens ont développé des algorithmes qui permettent, si pas de résoudre *exactement* un problème NP-complexe, de trouver une solution optimale qui sera garantie proche de la meilleure solution. Il s'agit de la famille des méta-heuristiques, des algorithmes stochastiques itératifs qui progressent vers un optimum global par échantillonnage de la fonction objectif. Il existe un grand nombre de méta-heuristiques différentes, allant de la simple recherche locale à des algorithmes complexes de recherche globale (Glover & Kochenberger 2003). Ces méthodes utilisent cependant un haut niveau d'abstraction, leur permettant d'être adaptées à une large gamme de problèmes différents. Nous nous pencherons donc sur différentes méta-heuristiques qui pourraient être adaptées au problème de l'inférence phylogénétique et qui permettraient de trouver une topologie optimale en un temps raisonnable pour de grandes phylogénies pouvant comporter plusieurs centaines de taxa et des milliers de caractères.

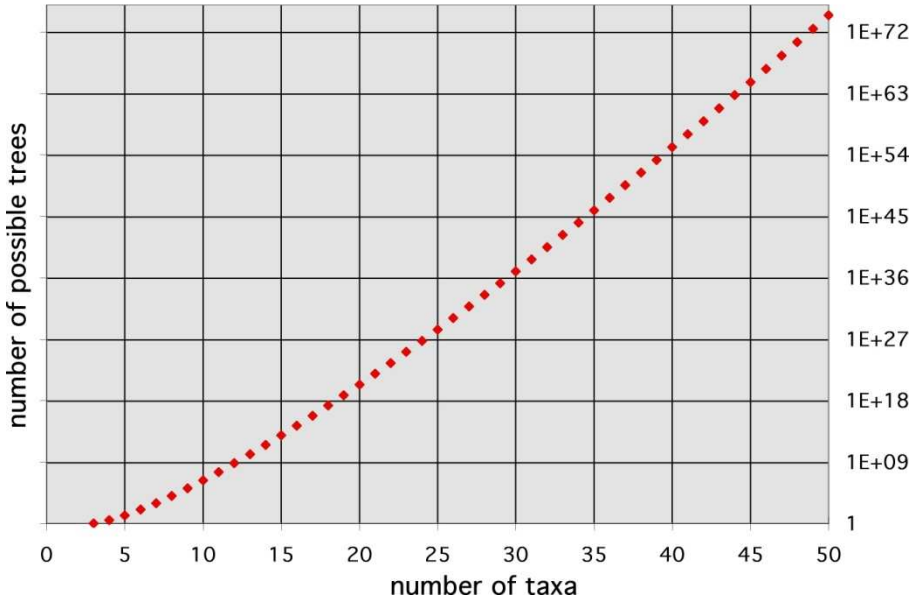

Figure 7 - Nombre de topologies possibles d'après le nombre de taxa.

LOGICIELS D'INFÉRENCE PHYLOGÉNÉTIQUES EXISTANTS

D'autres avant nous se sont bien évidemment penchés sur le problème de l'inférence phylogénétique, et nous allons ici décrire les logiciels les plus populaires basés sur le maximum de vraisemblance, en énonçant leurs caractéristiques principales.

PAUP* (*Swofford, 2003*). "Phylogeny Analysis Using Parsimony". Bien qu'implémentant au départ une méthode de parcimonie, il proposa par la suite des méthodes de distance et de maximum de vraisemblance (utilisant un Hill Climbing). Il implémente également un algorithme de quartet-puzzling. Ce fut, avec PHYLIP, le programme d'inférence phylogénétique le plus utilisé jusqu'au début des années 2000. Sa version Mac, la seule disposant d'une interface graphique, n'est cependant plus compatible avec Mac OS 10.5 et les récents

processeurs Intel. Il faut également noter que contrairement aux logiciels qui suivent, il n'est pas gratuit.

PHYLIP (*Felsenstein, 2005*). "PHYLogeny Inference Package". Un autre logiciel qui fut également très utilisé jusqu'à la fin des années 1990. C'est un ensemble de programmes permettant d'utiliser entre autre des méthodes de distances, de parcimonie et de maximum de vraisemblance, avec des algorithmes proches de ceux implémentés dans PAUP*.

Mr Bayes (*Huelsenbeck & Ronquist, 2001 & 2003*). Le logiciel d'inférence phylogénétique le plus utilisé actuellement. Il utilise une méthode Bayesienne, une Metropolis-coupled MCMC (MC^3). L'algorithme qu'il utilise va construire n chaînes de Markov en parallèle, dont n-1 sont « chauffées », en utilisant un paramètre de « chaleur » différent pour chacune. Plus une chaîne est chauffée, plus elle pourra facilement traverser les vallées de sa distribution de probabilités postérieures. Seule la chaîne non chauffée est utilisée pour estimer les probabilités postérieures, mais à chaque étape de l'algorithme, deux chaînes sont tirées au hasard et ont une probabilité d'échanger leur état (le dernier arbre accepté). De cette manière, la chaîne non chauffée explore localement la distribution des probabilités postérieures, et change régulièrement de région, grâce à l'échange de son état avec celui d'une chaîne qui explore des régions beaucoup plus vastes.

GARLI (*Zwickl, 2006*). "Genetic Algorithm for Rapid Likelihood Inference". Basé sur un algorithme génétique et s'inspirant de GAML (Lewis, 1998).

Tree-Puzzle (*Schmidt et al., 2002*). Basé sur l'algorithme du quartet-puzzling.

DAMBE (*Xia and Xie, 2001*). Utilise des méthodes de distance, de parcimonie et de maximum de vraisemblance. Pour le maximum de vraisemblance, il utilise l'algorithme DNAML.

PhyML (*Guindon & Gascuel, 2003*). Basé sur une heuristique de Hill Climbing, il a la particularité d'être également disponible via une interface web.

BAMBE (*Simon & Larget, 2000*). "Bayesian Analysis in Molecular Biology and Evolution". Il utilise une méthode Bayesienne, une variété d'algorithme de Metropolis-Hastings.

Metapiga_v1 (*Lemmon & Milinkovitch, 2002*). Utilise un algorithme génétique méta-populationnel original, le metaGA, basé sur le principe du « consensus pruning ».

En regardant la Table 1 ci-dessous, nous pouvons voir que peu de logiciels rassemblent l'ensemble des fonctionnalités qui y sont énoncées (ce qui correspondrait au logiciel « idéal »), et que seuls DAMBE, BAMBE et Metapiga_v1 proposent une interface utilisateur graphique. Beaucoup sont complexes à utiliser, et l'utilisateur doit avoir une bonne idée du modèle et des paramètres à utiliser avec ses données, ou utiliser un programme tiers pour les estimer. De plus, la moitié d'entre eux utilisent un format de fichier propre, plutôt qu'un standard extensible comme le format NEXUS (Maddison *et al.*, 1997). Seule la moitié d'entre eux gèrent un environnement multi-processeurs, et nécessitant pour chacun d'eux l'installation de la librairie MPI ; et aucun d'eux ne propose une version 64-bits pouvant tirer parti d'une machine où est installé plus de 4Go de mémoire RAM.

Table 1 - Comparaison des fonctionnalités des principaux logiciels d'inférence phylogénétique utilisant le maximum de vraisemblance.

Fonctionnalité	PAUP	PHYLIP	Mr Bayes	GARLI	Tree-Puzzle	DAMBE	PhyML	BAMBE	Metapiga_v 1	Idéal
Disponible sous Windows	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Disponible sous Mac OS X	*	☑	☑	☑	☑		☑		☑	☑
Disponible sous UNIX	☑	☑	☑	☑	☑	☑	☑	☑		☑
Disponible en version 64bits										☑
Gestion multi-processeur			☑	☑	☑		☑			☑
Ligne de commande / Interface texte	☑	☑	☑	☑	☑		☑	☑		☑
Interface graphique	*					☑		☑	☑	☑
Visualisation d'un arbre	*	☑				☑			☑	☑
Format de fichier propre pour les séquences		☑			☑		☑	☑		
Format NEXUS supporté	☑		☑	☑		☑			☑	☑
Modèles simples (JC, K2P)	☑	☑	☑	☑		☑	☑	☑	☑	☑
Modèles intermédiaires (HKY85, TN93)	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Modèle GTR			☑	☑	☑	☑	☑			☑
Hétérogénéité des taux par distribution gamma	☑	☑	☑	☑	☑	☑	☑			☑
Proportion de sites invariants	☑	☑	☑	☑	☑	☑	☑			☑
Évaluation du modèle le plus adapté						☑				☑
Optimisation des paramètres	☑		☑			☑			☑	☑
Gestion d'un outgroup	☑		☑			☑				☑
Partitioning des données			☑					☑		☑
Arbre de consensus	☑	☑	☑		☑		☑		☑	☑

* Disponible unique sous Mac OS inférieur à 10.5

Notre objectif sera donc d'implémenter un logiciel qui permettra l'estimation de grandes phylogénies en proposant un choix entre plusieurs méta-heuristiques, en regard des méthodes Bayesiennes déjà très bien supportées par Mr Bayes (*Huelsenbeck & Ronquist, 2001 & 2003*) ou BAMBE (*Simon & Larget, 2000*). Il tournera aussi bien sur les plateformes Windows, Mac OS X, qu'UNIX ; et pourra tirer parti des machines 64bits et/ou multiprocesseurs, qui deviennent standard de nos jours. Il pourra ainsi travailler sur des jeux de données beaucoup plus grands, n'étant plus limité par un adressage de mémoire 32-bits, et pouvant donc dépasser les 4Go. De même, l'usage de plusieurs processeurs ou cœurs pour paralléliser les calculs sera transparent pour l'utilisateur, sans installation de « packages » supplémentaires, et aussi simple qu'indiquer le nombre de processeurs à affecter à une analyse. Il proposera une interface graphique moderne et complète, avec une aide interactive pour guider l'utilisateur « débutant » dans ses choix, et dans l'estimation du modèle ou des paramètres à utiliser pour son jeu de données. Il tentera également d'intégrer dans un seul programme l'ensemble des fonctionnalités dans la Table 1, permettant à l'utilisateur d'avoir à sa disposition la plupart des outils de modélisation utilisés actuellement en phylogénie. Nous avons donc développé une version 2 de Metapiga, qui tente de s'approcher du logiciel « idéal » par les fonctionnalités qu'il propose.

ANCIENNES ET NOUVELLES MÉTA-HEURISTIQUES

Si le logiciel implémenté se voudra un cadre informatique le plus complet et convivial possible pour l'estimation de phylogénie, son moteur central sera la méta-heuristique qui recherchera l'arbre optimal. Il sera implémenté de telle manière qu'il ne repose pas sur une seule méta-heuristique, mais puisse proposer un nombre croissants de nouveaux algorithmes utilisant tous le même ensemble de modèles, d'estimateurs et d'outils communs à toutes les méta-heuristiques. Le logiciel pourra donc servir de « laboratoire *in silico* », dans lequel expérimenter une nouvelle heuristique sera beaucoup plus simple et rapide que d'implémenter un nouveau logiciel pour le faire.

Les outils communs à toutes les méta-heuristiques sont :

- Un estimateur, dans notre cas le maximum de vraisemblance.
- Un générateur de solution, c'est-à-dire une ou plusieurs méthodes pour générer un arbre de départ.
- Une boucle itérative qui répétera les différentes étapes de l'algorithme tout en disposant d'une ou plusieurs conditions d'arrêt.
- Des opérateurs de mutation permettant de perturber une solution. C'est grâce à eux que la méta-heuristique pourra parcourir l'espace de recherche, certaines utilisant des opérateurs locaux (perturbant peu une solution), globaux (perturbant énormément une solution), ou une combinaison des deux. Dans notre cas, nous devrons générer un ensemble d'opérateur permettant de perturber la topologie d'un arbre (localement ou globalement), mais également ses longueurs de branches ou les paramètres du modèle (taux de substitution, hétérogénéité des taux, etc.).

Une fois ces outils à notre disposition, nous aurons les briques nécessaires à la construction des méta-heuristiques que nous désirons implémenter. Nous avons sélectionné quatre méta-heuristiques, et aménagé le code pour pouvoir en implémenter d'autres facilement dans l'avenir. Les quatre candidats qui ont retenus notre attention sont le Hill Climbing (Guindon & Gascuel 2003, Mitchell & Holland 1993), le Simulated Annealing (Kirkpatrick *et al.* 1983), l'algorithme génétique (De Jong 1988) et le metaGA (*Lemmon & Milinkovitch* 2002).

HILL CLIMBING

L'algorithme de Hill Climbing est en fait une simple heuristique, car il ne dispose d'aucun moyen pour sortir d'un optimum local[21]. Il démarre d'un arbre aléatoire, et à chaque itération va perturber la solution courante en utilisant un opérateur de mutation (permettant de modifier la topologie de l'arbre ou les paramètres du modèle, voir « Opérateurs de mutation» page 48). Si la nouvelle solution est estimée meilleure, elle remplace la précédente ; sinon, elle est simplement éliminée. Comme l'algorithme ne fait que « monter » une pente dans l'espace de recherche, il sera naturellement bloqué au sommet d'un pic, et rien ne lui permettra de déterminer qu'il s'agit du pic le plus haut de l'espace de recherche. Il est donc très dépendant de la « topographie » (inconnue) de l'espace de recherche, ainsi que de la qualité de l'arbre de départ. Il est cependant rapide et simple à implémenter, et pourra servir de point de référence pour estimer l'efficacité des autres méta-heuristiques plus complexes. Il peut également être facilement étendu en « Greedy search » (appelé « algorithme glouton » en français, voir Figure 8), qui consiste simplement à répéter plusieurs Hill Climbing avec des solutions de départs aléatoires différentes, dans le but de visiter plusieurs pics de l'espace de recherche. Dans ce cas, seul le plus haut pic est gardé au final.

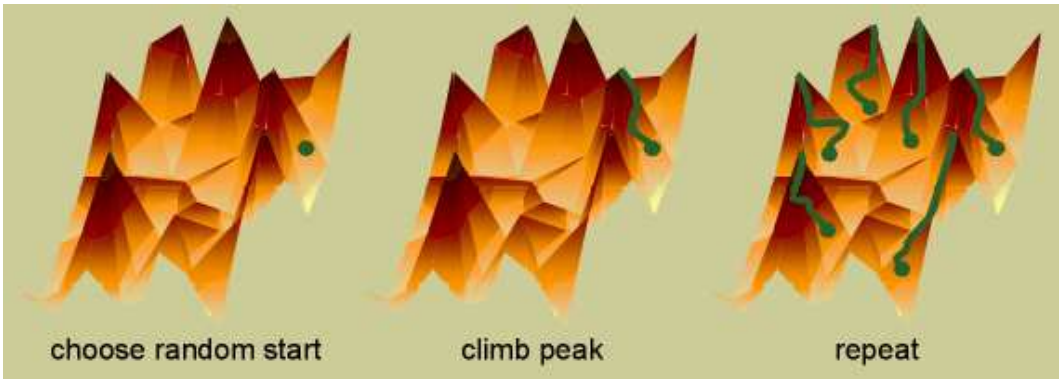

Figure 8 – Greedy search ("Algorithme Glouton"). Une solution de départ aléatoire est générée (le point vert dans l'espace de recherche), puis elle est améliorée jusqu'à atteindre le sommet d'un pic (optimum). Cette procédure est répétée plusieurs fois et seule la solution au sommet du plus haut pic rencontré est conservée.

[21] Optimum local : un pic dans l'espace de recherche des solutions, mais qui n'est pas le pic le plus haut (voir Figure 8). Le pic le plus haut de l'espace de recherche est l'optimum global, et il s'agit donc de la meilleure solution au problème.

SIMULATED ANNEALING

L'algorithme du Simulated Annealing (appelé « recuit simulé » en français) a été développé en 1983 par Kirkpatrick et Gelatt pour IBM comme un moyen de résoudre des problèmes complexes d'optimisation. Il est basé sur un principe physique : lorsqu'un solide est chauffé jusqu'à son point de fusion, puis lentement refroidi, les molécules de ce solide s'alignent naturellement dans une formation d'énergie minimale. Le Simulated Annealing reproduit ce processus en considérant chaque état dans un espace de recherche comme ayant une énergie (proportionnelle à la valeur de vraisemblance de cet état) et en tentant de trouver un état avec une énergie globalement minimale (c'est-à-dire la solution optimale).

Le Simulated Annealing démarre avec un état initial dans l'espace de recherche (l'arbre de départ) et crée ensuite un état aléatoire dans un voisinage donné de l'état initial (grâce aux opérateurs de mutation). Si le nouvel état est une meilleure solution (énergie inférieure, meilleure valeur de vraisemblance), l'algorithme choisit le nouvel état comme état courant. Si le nouvel état est une solution moins bonne (énergie supérieure, valeur de vraisemblance moins bonne), le nouvel état est accepté comme état courant selon une probabilité égale à la probabilité de Boltzmann : $e^{\Delta E/T}$, où ΔE est la différence en énergie des deux états (ΔE < 0) et T est la température du système. Si la température T est diminuée suffisamment lentement, l'algorithme garanti de trouver un état optimal (énergie minimale) (Kirkpatrick *et al.* 1983).

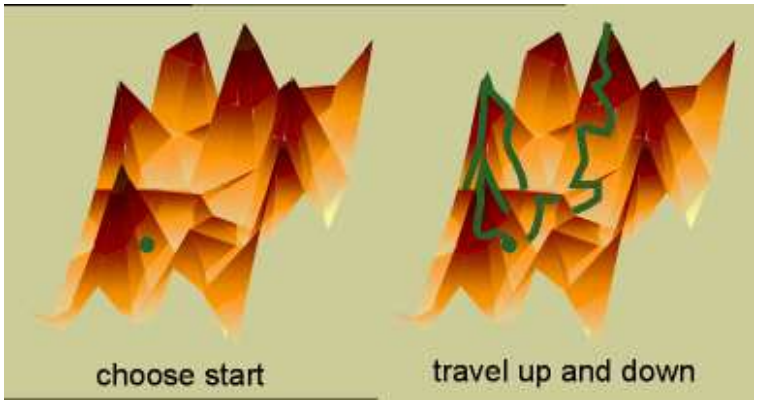

Figure 9 – Simulated Annealing ("Recuit Simulé"). Contrairement à l'algorithme glouton qui nécessite de visiter chaque pic de l'espace de recherche pour trouver le plus haut (vu qu'il reste bloqué au sommet), cet algorithme permet de voyager dans l'entièreté de l'espace de recherche, sans jamais rester bloqué.

Grâce à cette capacité d'accepter momentanément un état sous-optimal, le Simulated Annealing est un algorithme idéal pour résoudre des problèmes NP-complexes qui ont souvent une multitude d'optima locaux dans leur espace de recherche, la nature stochastique du Simulated Annealing permettant de s'échapper de ces optima locaux en considérant que la température est diminuée suffisamment lentement. La fonction qui détermine à quelle vitesse un système est refroidi dans le Simulated Annealing est appelée « courbe de refroidissement » (« cooling schedule »). Comme refroidir le système trop rapidement peut « étouffer » le système et générer des résultats sous-optimaux, la courbe de refroidissement a une

importance capitale dans le succès de l'algorithme. Trouver une courbe de refroidissement adaptée dépendant beaucoup du jeu de données, nous implémenterons une quinzaine de courbes différentes, en incluant celle décrite par Lundy (Lundy 1985) qui prend en compte certaines données biologiques, telles que le nombre de séquences, le nombre de sites et la vraisemblance du Neighbor Joining Tree.

Nous implémenterons également divers paramètres qui pourraient avoir un impact important sur les performances de l'algorithme : le calcul de la température de départ, la probabilité d'acceptance maximale, la fréquence à laquelle la température est diminuée et la possibilité de réchauffer la température sous certaines conditions.

ALGORITHME GÉNÉTIQUE

L'algorithme génétique est une méta-heuristique inspirée par la biologie évolutionniste, qui reproduit certains mécanismes tels que mutation, sélection et recombinaison (De Jong 1988). Il travaille avec une population évolutive d'individus, représentée par un ensemble d'arbres de départ. À chaque itération de l'algorithme (appelées génération), chaque individu subit une mutation (un opérateur de mutation est utilisé sur chaque arbre). Ensuite, une phase de sélection a lieu, en suivant un schéma défini (plusieurs stratégies de sélection seront implémentées). Le résultat obtenu est une nouvelle population, où les individus faibles (les arbres avec une mauvaise valeur de vraisemblance) ont une certaine probabilité d'être éliminés et remplacés par des copies des individus forts (les arbres avec de bonnes valeurs de vraisemblance). Durant une sélection, il peut aussi survenir un événement de recombinaison, chaque individu faible ayant une chance de se recombiner avec un individu fort (plutôt que d'être complètement remplacé par une copie du fort). En pratique, un arbre phylogénétique peut se recombiner avec un autre s'ils ont une branche en commun, partitionnant l'ensemble des taxa de manière identique mais ayant potentiellement des sous-topologies différentes. La recombinaison consiste à échanger les deux sous-arbres descendant de cette branche commune.

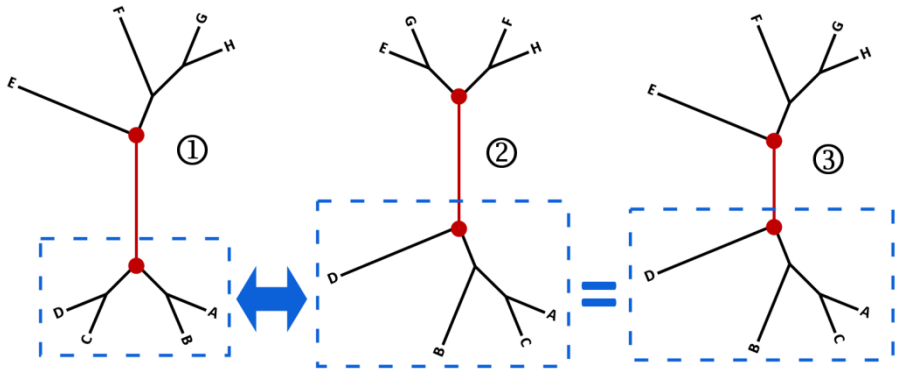

Figure 10 – Recombinaison entre 2 arbres phylogénétiques. L'arbre ① peut se recombiner avec l'arbre ②, car ils possèdent au moins une branche en commun. Nous avons choisi la branche rouge, qui divise l'arbre en formant la bipartition [A,B,C,D] vs [E,F ,G,H]. La partition [A,B,C,D] ayant une topologie différente dans ① et ②, nous obtenons une nouvelle topologie (l'arbre ③) après recombinaison de ① avec ②.

METAGA

Il s'agit d'un algorithme génétique méta-populationnel, développé par Lemmon et Milinkovitch en 2002, que nous allons intégrer tout en y ajoutant certains nouveaux paramètres. Le metaGA se base sur les mêmes mécanismes qu'un algorithme génétique classique, mais exploite deux populations ou plus, coexistant en parallèle et interagissant dans un cadre méta-populationnel. Comme le metaGA implique plusieurs recherches en parallèle, une grande variété inter-populationnelle peut être maintenue, même lorsque chaque population subit une forte pression sélective. Toutefois, l'essence même du metaGA implique que les populations ne soient pas totalement indépendantes, car forcées de coopérer dans la recherche de l'arbre optimal. Au sein de chaque population, les arbres sont soumis à des événements d'évaluation, de sélection, de recombinaison, et de mutation comme ils le seraient dans un algorithme génétique à une seule population, mais tous les opérateurs de mutation topologiques sont guidés par des comparaisons inter-populationnelles. Le metaGA est basé sur l'hypothèse que ces comparaisons conduisent à l'identification de partitions qui sont correctes (et ne devraient donc pas être modifiées) et de régions devant toujours être résolues. Le « consensus pruning » détermine quelles branches ne devraient pas être mutées, fixant définitivement la partition, en suivant deux schémas (consensus stricts ou stochastiques). Avec des consensus stricts, tous les individus doivent posséder une partition en commun pour empêcher une mutation de l'affecter. Avec des consensus stochastiques, chaque partition peut empêcher une mutation avec une probabilité proportionnelle au nombre d'individus possédant cette partition.

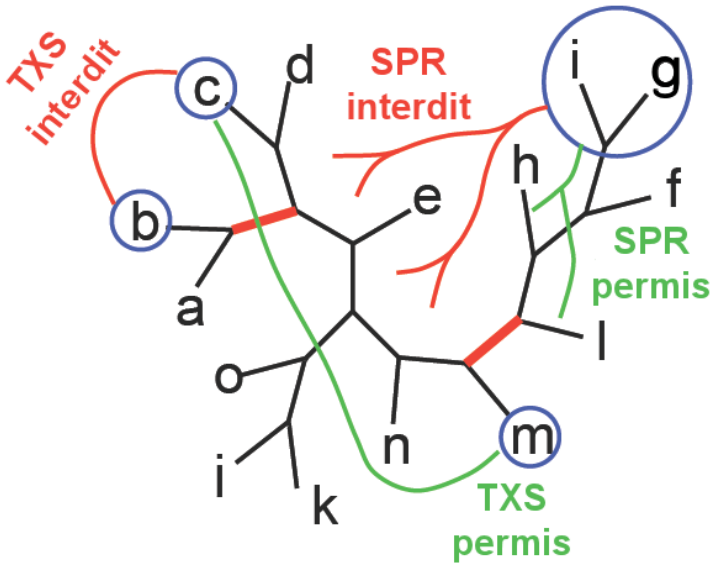

Figure 11 – Consensus Pruning. (Lemmon & Milinkovitch 2002) Dans cet exemple, supposons que l'intégralité des arbres de chaque population possèdent les partitions [a,b] et [f,g,h,i,l] en commun (les 2 branches rouges). Lors de la prochaine phase de mutation, les mutations de l'arbre qui brisent l'un de ces consensus seront interdites (traits rouges). Le fonctionnement des opérateurs de mutation est décrit en détail dans la section « Opérateurs de mutation» page 48.

Étant donné que fixer une partition pour toujours pourrait amener l'algorithme à rester bloqué dans un optimum local, nous allons ajouter un paramètre de tolérance qui conférera toujours une petite chance à l'algorithme de briser une partition qui aurait été fixée par consensus. Des partitions définitivement fixées par consensus nous conduisent aussi à prendre certaines décisions sur le comportement des opérateurs de mutations. Nous implémenterons deux schémas de décision : le premier rendra les opérateurs « aveugles », annulant simplement une mutation lorsqu'elle devrait briser un consensus ; et le deuxième rendra les opérateurs « supervisés », sélectionnant les cibles des opérateurs de mutation parmi un ensemble de candidats acceptables (ceux qui ne briseront aucun consensus). Ces deux politiques pourront avoir des effets différents au début ou à la fin de l'algorithme, un opérateur aveugle étant efficace en début de recherche (peu ou pas de consensus empêchant un opérateur de fonctionner), mais pouvant devenir inefficace en fin de recherche (avec beaucoup de consensus, les chances de ne pas les affecter diminuent). A l'inverse un opérateur supervisé permettra de rapidement terminer une recherche vu le nombre restreint de candidats potentiels en fin de recherche, mais sera très gourmand en temps de calcul en début de recherche, le nombre de candidats potentiels à évaluer pouvant être très grand. Enfin, comme la recombinaison existe au sein d'une population, nous avons également imaginé une recombinaison inter-population : à chaque génération, il y a une chance que plutôt qu'être mutés, tous les individus d'une population choisie au hasard soient recombinés avec des individus provenant exclusivement des autres populations (avant mutation).

MÉTHODOLOGIE ET CONCEPTION DU LOGICIEL METAPIGA 2.0

Nous avons défini le cadre informatique que nous allons mettre en place, et nous allons l'implémenter sous la forme d'un logiciel : « MetaPIGA 2.0 ». Nous ne reprendrons de la première version du logiciel Metapiga (Lemmon & Milinkovitch, 2002) que le concept de son heuristique, le metaGA ; MetaPIGA 2.0 sera bien un logiciel original. Par la suite, nous utiliserons indifféremment les termes « MetaPIGA 2.0 » ou simplement « MetaPIGA » pour nous référer au logiciel que nous allons développer. Nous détaillerons dans cette section nos choix d'implémentation, quels outils et modèles sont à disposition de l'utilisateur ainsi que certaines optimisations apportées au calcul de la vraisemblance.

LANGAGE DE PROGRAMMATION UTILISÉ

Pour le langage de programmation, notre choix s'est posé sur le JAVA 6. Ce langage a l'avantage de tourner sur une machine virtuelle (JVM), permettant au code d'être uniquement dépendant ce cette JVM, et pas de la machine sur laquelle le logiciel tournera. Il permettra donc au même code (optimisé pour une seule JVM) de fonctionner aussi bien sur une machine Windows, Mac OS X ou UNIX (ou toute autre machine pour laquelle une JVM aura été développée). Une critique qui a souvent été formulée envers le JAVA est sa lenteur, due au fait qu'il passait par une machine virtuelle, comparativement à un langage compilé pour une

machine spécifique (comme le C/C++) (Reinholtz 2000). Si cette remarque était effectivement justifiée il y a quelques années (JAVA 4 et ses prédécesseurs), la machine virtuelle actuelle (JAVA 6) donne d'excellentes performances, qui se rapprochent suffisamment de celles d'un langage comme le C/C++ pour nous permettre de l'utiliser dans un projet qui nécessite des calculs intensifs. De fait, JAVA nous offrira les avantages bien connus d'avoir à beaucoup moins gérer la mémoire que dans un langage comme le C/C++ (grâce au « Garbage collector » de la JVM) et de disposer d'excellents outils pour générer une interface utilisateur à la fois conviviale et efficace.

Un autre gros avantage que nous procurera JAVA est le fait qu'il est nativement parallélisable, via le « multithreading ». En effet, pour faire tourner des tâches en parallèle en JAVA, il suffit de générer et de lancer autant de « threads[22] », et c'est l'implémentation de la machine virtuelle qui se chargera de les assigner aux processeurs / cœurs disponibles (nous évitant de devoir développer des stratégies de distribution, qui seraient de plus spécifiques pour chaque type de machine). Nous pourrons donc très facilement proposer à l'utilisateur de faire tourner en parallèle ce qui peut l'être, sans qu'il ait à installer de librairie supplémentaire, comme la MPI nécessaire à la parallélisation en C/C++ par exemple.

Autre avantage de JAVA, une version 64-bit de la JVM existe maintenant pour toutes les plateformes qui nous intéressent (Windows, Mac OS X et UNIX), ce qui nous permettra d'avoir une version de MetaPIGA qui puisse disposer de plus de 2Go de mémoire RAM (le maximum qu'on puisse normalement utiliser avec une JVM 32-bit). Les arbres, et surtout le calcul de la vraisemblance, sollicitent une place importante en mémoire, et certaines méta-heuristiques nécessitent de travailler sur de nombreux arbres en même temps. De plus, si nous voulons paralléliser plusieurs analyses pour profiter d'une machine multiprocesseurs, il faut multiplier d'autant la mémoire nécessaire pour faire tourner une analyse. Enfin, notre but étant de pouvoir travailler avec des grandes phylogénies de plusieurs centaines de taxa et plusieurs milliers de nucléotides, la taille de ces structures va très vite atteindre des proportions de l'ordre du méga-octet, et une analyse (et surtout plusieurs analyses en parallèle) avec un tel jeu de données nécessitera plusieurs giga-octets de mémoire.

Le choix de JAVA 6 comme langage de programmation nous semble donc particulièrement bien adapté à la conception de notre projet.

[22] **Thread** : « unité d'exécution », c'est-à-dire composante d'un processus, correspondant à une instruction élémentaire effectuée dans le programme, et qui appartient à un seul processus. Lors de l'exécution du processus, le processeur partage son temps de traitement entre les fils appartenant à ce processus.

FONCTIONNALITÉS IMPLÉMENTÉES

De manière générale, nous voulons que MetaPIGA soit un logiciel ergonomique, convivial et relativement simple d'utilisation. Par simple d'utilisation, nous voulons dire qu'il ne nécessite pas de lire 100 pages d'un manuel avant de pouvoir être utilisé, et qu'il puisse estimer une phylogénie correctement avec un paramétrage par défaut sans que l'utilisateur ait à gérer lui-même des dizaines de paramètres. Malgré tout, nous voulons également qu'un utilisateur expert avec les outils de modélisation existants en phylogénie puisse paramétrer avec précision son analyse. Pour atteindre ce but, nous proposons à la fois une interface graphique et un mode « ligne de commande », tout en utilisant des formats de fichier standard.

MetaPIGA utilise la norme NEXUS (Maddison *et al.* 1997), qui a l'avantage d'être facilement extensible pour répondre aux besoins d'un logiciel, mais de manière transparente pour les autres. La norme NEXUS fonctionne par blocs de données, et définit une série de blocs standard que tous les logiciels prenant en charge cette norme sont censés pouvoir décoder. Chaque logiciel peut ensuite ajouter autant d'autres blocs spécifiques qu'il le désire, qui ne parasiteront pas les blocs standards, permettant aux autres logiciels de toujours pouvoir lire ces fichiers. MetaPIGA n'utilise que les informations des NEXUS blocs CHARACTERS, DATA et TREES, contenant l'alignement de séquences, les informations qui lui sont relatives et potentiellement des arbres fournis par l'utilisateur. En plus de ces derniers, nous avons décrit un bloc METAPIGA, qui contient tous les paramétrages spécifiques à MetaPIGA. Sauvegarder la configuration de MetaPIGA pour une analyse se fait donc sous la forme d'un fichier NEXUS, qui pourra être lu sans modification dans beaucoup d'autres logiciels de phylogénie qui utilisent eux aussi ce standard. Les arbres qui entrent ou qui sortent de MetaPIGA seront également sous un format NEXUS (dans un fichier TREE), et interprétés par la plupart des autres logiciels qui comprennent le format Newick.

Les fichiers NEXUS nous permettent d'utiliser aisément la version « ligne de commande » de MetaPIGA, qui prend simplement un fichier NEXUS en paramètre : il contient les séquences, les arbres éventuels, et un bloc METAPIGA avec le paramétrage complet de l'analyse à effectuer. Comme il existe une valeur par défaut pour chaque paramètre, un fichier NEXUS sans bloc METAPIGA sera également bien interprété et permettra une analyse avec les paramètres par défaut. La version ligne de commande permet donc de préparer un paramétrage à l'avance et d'être utilisée facilement depuis un script ou un autre programme, ou simplement sur une machine ne disposant pas d'un environnement graphique.

Pour une utilisation plus standard, l'interface graphique de MetaPIGA permet de paramétrer une analyse via une suite logique de panneaux de configuration, qui disposent d'une aide interactive, détaillant chaque paramètre dans une fenêtre d'aide commune, en passant simplement la souris au-dessus d'un paramètre. Les panneaux de configuration se succèdent dans une suite logique, regroupant les paramètres de même nature. L'utilisateur commence par le premier onglet permettant de paramétrer son jeu données (sélection d'un outgroup ou partitionnement du jeu de données), puis le deuxième onglet où il choisi la méta-heuristique (et son paramétrage) qui effectuera l'analyse. Le troisième onglet lui permet de sélectionner le modèle d'évaluation de la vraisemblance, et la manière d'optimiser ses paramètres pendant l'analyse. Des outils d'aide à la décision sont accessibles ici par simple clic sur un bouton,

permettant d'évaluer les paramètres de départ du modèle, ou de déterminer automatiquement quel modèle est le mieux adapté pour le jeu de donnée de l'utilisateur (via des méthodes comme le « Likelihood Ratio Test »). Ensuite le quatrième onglet permet à l'utilisateur de choisir la méthode avec laquelle le ou les arbres de départ seront générés. Le cinquième onglet proposera les opérateurs de mutation que l'heuristique peut utiliser selon les paramètres déjà entrés, avec la possibilité de fournir la fréquence d'utilisation de chacun d'eux. Une simple case à cocher pour chacun d'eux permet de rendre l'opérateur dynamique, permettant alors à l'heuristique d'ajuster régulièrement la fréquence d'utilisation de cet opérateur selon ses résultats actuels (les fréquences seront ajustées pour être proportionnelles au gain de vraisemblance apporté par les opérateurs depuis le dernier ajustement). Enfin, le dernier onglet permet de définir les conditions d'arrêt de l'heuristique, le nombre de réplicats à produire et les données d'output à générer. La encore, 2 outils permettent d'avoir un paramétrage automatique sur la condition d'arrêt et le nombre de réplicats nécessaires. Une condition d'arrêt automatique est disponible pour toutes les heuristiques, stoppant l'analyse lorsque la valeur de vraisemblance ne s'améliore pas plus de 0.01% durant un certain nombre d'itération. Une condition d'arrêt automatique spécifique au metaGA est aussi proposée, nous la décrirons en détails dans la section Heuristiques (page 38). Pour les réplicats, le paramétrage automatique permet à MetaPIGA de stopper la production de réplicats quand l'erreur moyenne relative des valeurs de support de l'arbre de consensus est en dessous d'un certain seuil.

Nous allons maintenant parcourir plus en détails les différents paramètres implémentés, et les outils satellites permettant de faciliter l'utilisation du logiciel.

FONCTIONNALITÉS DE MANIPULATION DU JEU DE DONNÉES

MetaPIGA permet à l'utilisateur de facilement exclure des taxa ou des sites de son jeu de données, sans affecter son fichier de données original. De plus, il lui permet de définir un ou plusieurs taxa comme faisant partie de l'outgroup. Ces séquences seront identifiées comme extérieures au groupe monophylétique[23] étudié. Pour bien faire, l'outgroup devrait être choisi de manière à ce qu'il soit suffisamment proche des autres espèces, tout en étant certain qu'il soit extérieur au groupe monophylétique étudié. Par exemple, pour un jeu de données contenant des séquences d'humains et de chimpanzés, les gorilles sont un bon choix d'outgroup. Les taxas ne faisant pas partie de l'outgroup font automatiquement partie de ce que nous appelons l'ingroup. MetaPIGA racinera toujours l'arbre en sélectionnant l'outgroup (s'il l'utilisateur en a construit un) comme l'un des trois descendants du nœud racine. De plus, aucun opérateur de mutation topologique ne pourra casser la partition « outgroup vs ingroup », dans le cas où l'outgroup serait composé de plusieurs taxa.

Une autre fonction permettant de manipuler le jeu de données est la possibilité de le partitionner à l'aide de « charsets », un regroupement de sites (ou caractères). Un outil graphique permet à l'utilisateur de facilement créer des charsets, en sélectionnant les sites qui en feront partie, ou en entrant un intervalle de positions. Ensuite, l'utilisateur peut

[23] **Monophylétique** : un groupe incluant un ancêtre commun et la totalité de ses descendants.

sélectionner un ou plusieurs charsets pour partitionner l'entièreté du jeu de données. MetaPIGA suppose que chaque partition évolue avec la même topologie, mais que tous les autres paramètres (fréquences des bases à l'équilibre, matrice de taux, forme de la distribution gamma utilisée pour l'hétérogénéité des taux, et proportion de sites invariants) sont estimés et optimisés séparément pour chaque partition. De plus, les longueurs de branche relatives sont estimées conjointement, mais l'apport absolu à l'arbre pour chaque partition est estimé séparément. Nous avons donc introduit un paramètre supplémentaire dans l'équation de la vraisemblance, la variation de taux parmi les partitions (« among-partition rate variation »), un facteur qui modifie la longueur de branche au sein de chaque partition. Les longueurs de branches relatives sont optimisées normalement, mais le paramètre additionnel pour chaque partition représente le taux relatif et est optimisé (et muté) séparément. Si Θ_p représente ce taux de variation relatif pour la partition p, il peut être incorporé dans le calcul de la vraisemblance en multipliant systématiquement t par Θ_p dans les équations permettant de calculer la matrice de transition (Eq.11, Eq.13, Eq.15 et Eq.19). Le taux relatif Θ_p est contraint d'avoir une moyenne pondérée de 1, la pondération étant proportionnelle au nombre de sites dans la partition. Chaque paramètre Θ_p doit donc être ajusté de telle manière que :

$$\sum_{p}^{nPar} S(p).\Theta_p = 1 \qquad \textbf{Eq.25}$$

Où $nPar$ est le nombre de partitions divisant le jeu de données, et $S(p)$ est le nombre de sites contenus dans la partition p.

HEURISTIQUES

Nous avons déjà décrit le concept des différentes méta-heuristiques dans l'introduction de ce chapitre (Anciennes et nouvelles méta-heuristiques page 29). Nous allons détailler ici les différents paramètres permettant d'ajuster avec précision leur comportement. Nous ne parlerons pas du Hill Climbing qui n'a aucun paramétrage associé.

SIMULATED ANNEALING

Le Simulated Annealing démarre d'un arbre unique, et explore l'espace de recherche en effectuant des perturbations locales sur cet arbre (topologiques ou relatives aux paramètres du modèle, selon les opérateurs qui seront choisis). Les arbres ainsi obtenus ayant une meilleure vraisemblance sont toujours acceptés, tandis que les arbres ayant une vraisemblance inférieure sont acceptés avec une probabilité A_i qui est à la fois fonction de la perte relative de vraisemblance et d'un paramètre de contrôle appelé température. Tous les paramètres proposés par MetaPIGA permettent d'ajuster cette température d'une manière ou

d'une autre. Le paramètre le plus important est la forme de la courbe de refroidissement, qui indique à l'algorithme de quelle manière réduire la température à chaque décrémentation. Quatorze courbes différentes sont proposées, où T_i est la température après i décrémentations et Γ est le nombre maximum de décrémentation de température avant de la réinitialiser à la température de départ T_0 (grâce au paramètre de réchauffement décrit ci-dessous). Mis à part pour la courbe de refroidissement de Lundy, T_0 (et T_Γ lorsqu'il s'applique) est calculé de cette manière :

$$T_0 = \left|\frac{-\Delta L}{\ln \mathrm{A}_0}\right| \qquad \text{et} \qquad T_\Gamma = \left|\frac{-\Delta L}{\ln \mathrm{A}_\Gamma}\right| \qquad \text{Eq.26}$$

Où ΔL est une limite supérieure sur le changement de la vraisemblance, A_0 est le paramètre d'acceptance initiale et A_Γ le paramètre d'acceptance finale. L'acceptance initiale est la probabilité maximale initiale d'accepter un arbre avec une mauvaise vraisemblance, définissant donc la température de départ utilisée quand le Simulated Annealing démarre ou lorsque la température est réinitialisée. L'acceptance finale est la probabilité maximale finale d'accepter un arbre avec une mauvaise vraisemblance, définissant donc la température finale utilisée quand le Simulated Annealing devrait se terminer ou juste avant de réinitialiser la température. Les courbes de refroidissement proposées sont reprises dans la Table 2, et l'utilisateur peut donner en paramètre les valeurs de A_0 et A_Γ pour les courbes qui les utilisent.

Si la courbe de refroidissement décrit de quelle manière la température est diminuée, un autre paramètre spécifie après combien d'itérations du Simulated Annealing cette décrémentation a lieu (donc après combien de modifications de l'arbre). En plus de pouvoir spécifier un simple nombre d'itérations, il est possible de spécifier que la température doit être diminuée après *s* succès ou *f* échecs, selon ce qui se produit en premier. Les succès sont des modifications de l'arbre qui améliorent la vraisemblance et les échecs sont celles qui ne l'améliorent pas. Il est également possible de spécifier une condition pour laquelle la température doit être réinitialisée, soit après un certain nombre d'itérations de l'algorithme, soit lorsque la température atteint un seuil minimal (exprimé comme un pourcentage de la température de départ). Enfin, il est possible de déterminer comment ΔL est initialisé. ΔL est utilisé pour calculer la température de départ, et est la distance maximale entre une solution courante et la pire solution qui puisse être acceptée avec une probabilité de A_i. ΔL peut être initialisé à un pourcentage *p* de la vraisemblance du Neighbor Joining Tree, ou estimé suite à une période de « burn-in » durant laquelle les opérateurs de mutation sélectionnés sont utilisés sur l'arbre de départ pour un total de 20 applications chacun. L'écart de vraisemblance maximal observé durant cette période est alors utilisé comme ΔL.

Table 2 - Courbes de refroidissement du Simulated Annealing

Type de courbe	Équation
Lundy (avec c et α donnés en paramètre)	$T_{i+1} = \frac{\Delta L}{1 + i\beta}$ Avec $\beta = \frac{c}{(1-\alpha)n + \alpha^{\frac{-\ln NJT}{m}}}$ qui est le paramètre qui contrôle le taux de refroidissement (sa valeur est < 1) où n est le nombre de séquences, m est le nombre de sites, c et α ont une valeur entre 0 et 1 et $\ln NJT$ est le log de la vraisemblance de l'arbre construit par Neighbor Joining.
Ratio-Percent (de paramètre δ)	$T_{i+1} = \delta . T_i$ avec $\delta < 1$
Fast Cauchy	$T_i = \frac{T_0}{i}$
Boltzmann	$T_i = \frac{T_0}{\ln i}$
Géométrique (de paramètre α)	$T_i = T_0 . \alpha^i$ avec $\alpha < 1$
Linéaire	$T_i = T_0 - i\frac{(T_0 - T_\Gamma)}{\Gamma}$
Triangulaire	$T_i = T_0 . (\frac{T_0}{T_\Gamma})^{\frac{i}{\Gamma}}$
Polynomiale	$T_i = \frac{(T_0 - T_\Gamma)(\Gamma + 1)}{\Gamma(i+1)} + T_0 - \frac{(T_0 - T_\Gamma)(\Gamma + 1)}{\Gamma}$
Transcendantale (exponentielle)	$T_i = T_\Gamma + \frac{(T_0 - T_\Gamma)}{1 + e^{3(i - \frac{\Gamma}{2})}}$
Transcendantale (logarithmique)	$T_i = T_0 . e^{-(\frac{i}{\Gamma})^2 \ln \frac{T_0}{T_\Gamma}}$
Transcendantale (périodique)	$T_i = \frac{T_0 - T_\Gamma}{2} . \left(1 + \cos i\frac{\Pi}{\Gamma}\right) + T_\Gamma$
Transcendantale (périodique lissée)	$T_i = \frac{T_0 - T_\Gamma}{4} . \left(2 + \cos 8i\frac{\Pi}{\Gamma}\right) . e^{-\frac{i}{2\Gamma}}$
Hyperbolique (tangente)	$T_i = \frac{T_0 - T_\Gamma}{2} . \left(1 - \tanh(\frac{10i}{\Gamma} - 5)\right) + T_\Gamma$
Hyperbolique (cosinus)	$T_i = \frac{T_0 - T_\Gamma}{\cosh \frac{10i}{5}} + T_\Gamma$

ALGORITHME GÉNÉTIQUE

L'algorithme génétique utilise un ensemble (« population ») d'arbres (« individus ») de départ, dont la taille est choisie par l'utilisateur. A chaque itération de l'algorithme (« génération »), la population subit une phase de mutation suivie d'une phase de sélection. Lors de la phase de mutation, chaque individu est muté via un opérateur, excepté le meilleur individu de la génération précédente (l'arbre ayant la meilleure vraisemblance). Les opérateurs de mutation sont choisis parmi ceux sélectionnés par l'utilisateur, et il peut configurer l'heuristique pour qu'elle utilise le même opérateur pour toute une population, ou un opérateur différent pour chaque individu. La phase de sélection permet de déterminer quels seront les individus qui pourront générer une descendance, qui constituera la génération suivante. Pour un arbre phylogénétique, générer une descendance s'effectue en copiant simplement cet arbre à l'identique. Plusieurs stratégies sélectives ont été implémentées, offrant une diversité de pression sélective. Elles sont détaillées dans la Table 3 ci-dessous.

Table 3 – Types de sélections disponibles pour l'algorithme génétique

Type de sélection	Description	Pression sélective
Rank	On associe à chaque individu une probabilité de générer une descendance proportionnelle à leur position dans une liste dans laquelle ils sont classés par leur score (vraisemblance). Les individus qui ne génèrent pas de descendance sont remplacés par une copie du meilleur individu.	Moyenne
Tournament	Deux individus sont choisis aléatoirement dans la population de n individus, et une descendance est produite par celui des deux qui a le meilleur score. Les deux individus sont ensuite réinsérés dans cette population, et le processus est répété jusqu'à ce que n descendants aient été produits.	Faible
Replacement	Deux individus sont choisis aléatoirement dans la population de n individus, et deux copies de celui ayant le meilleur score sont réinsérées dans cette population (les « parents » sont écartés). Le processus est répété sn fois, où s est la force de la sélection (un paramètre défini par l'utilisateur), puis une copie de la population obtenue après cette sélection est utilisée comme descendance.	Variable
Improve	Seuls les individus ayant un meilleur score que celui du meilleur individu de la génération précédente sont gardés. Chaque individu ratant ce test est écarté et remplacé par une copie du meilleur individu de la génération actuelle.	Forte
Keep the best	Seul le meilleur individu de la génération est gardé, tous les autres étant remplacé par une copie de celui-ci.	Très forte

Lors de la phase de sélection, nous avons également introduit le concept de recombinaison. La descendance de deux individus est normalement générée en éliminant l'individu le plus faible et en copiant l'individu le plus fort (en suivant la stratégie sélective choisie). Mais selon le taux de recombinaison entré par l'utilisateur, il y a une chance pour chaque individu éliminé qu'il soit remplacé par une descendance qui est une recombinaison de lui-même avec l'individu le plus fort. En pratique, une branche (partition) commune aux deux arbres est échangée, en gardant de préférence la partition la plus grande appartenant à l'individu le plus fort. Si aucune branche n'est commune aux deux individus, la descendance reste une simple copie de l'individu le plus fort.

METAGA

Basé sur l'algorithme génétique, le metaGA utilise plusieurs populations d'individus forcées de coopérer entre elles dans la recherche de l'arbre optimal (L'utilisateur fixe le nombre de populations et d'individus dans chacune d'elles). Les deux mêmes phases de mutation et de sélection présentes dans l'algorithme génétique sont également présentes dans le metaGA, et les mêmes stratégies de sélection sont disponibles. La grande différence dans la phase de mutation est que les opérateurs de mutation topologique doivent obéir aux limites fixées par le « consensus pruning ». Avant chaque phase de mutation, une liste de consensus est construite en utilisant les branches communes à plusieurs arbres. Deux types de consensus peuvent être choisis. Le consensus strict, qui nécessite qu'une branche soit commune à la totalité des individus (dans toutes les populations) pour faire partie du consensus ; et le consensus stochastique, qui ajoute une branche au consensus dès qu'elle est commune à au moins deux arbres, en sauvegardant le nombre d'arbres possédant cette branche. Pour qu'une branche soit commune entre plusieurs arbres, il faut qu'elle génère la même bipartition de taxa, composée des 2 groupes de taxa se trouvant de part et d'autre de cette branche (voir Figure 12 ci-dessous).

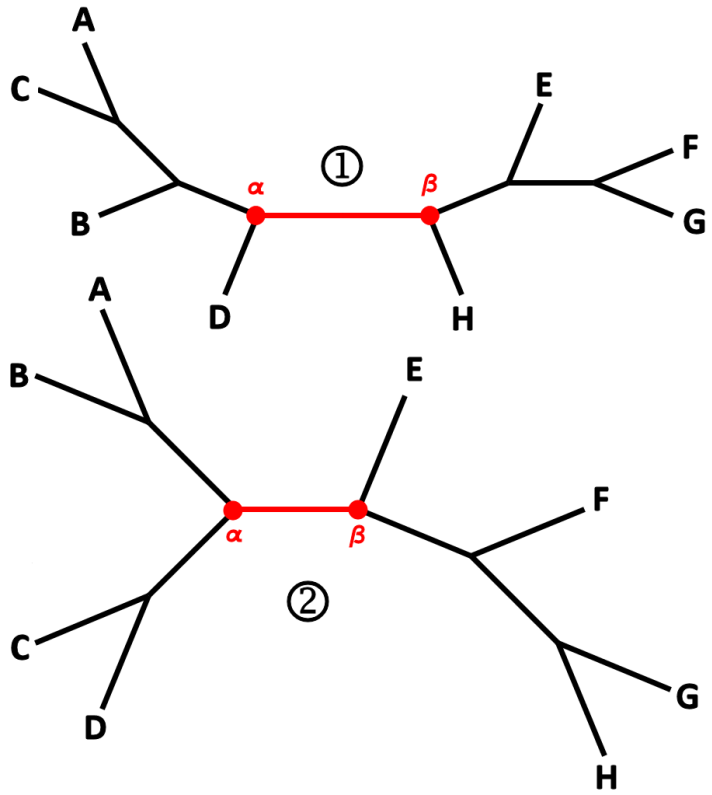

Figure 12 – Dans ces 2 arbres, la branche reliant le nœud α au nœud β divise l'arbre en 2 partitions, la première formée des taxa {A,B,C,D} et la seconde formée des taxa {E,F,G,H}. La bipartition générée par cette branche $\alpha - \beta$ est donc {A,B,C,D} versus {E,F,G,H}. La branche $\alpha - \beta$ est donc comme à ces 2 arbres, bien que leur topologie soient différentes (par exemple, la partition {A,C} existe dans l'arbre ① et pas dans l'arbre ②).

Lors d'une mutation topologique, l'opérateur vérifie que la mutation ne brise aucun consensus présent dans cette liste. Il empêchera systématiquement une mutation brisant un consensus en cas de consensus strict, et avec une probabilité proportionnelle au nombre d'arbres possédant le consensus dans le cas de consensus stochastique. Par exemple, en considérant 4 populations de 4 individus et des consensus stochastique, une branche commune à 12 arbres sur les 16 aura 75% de chances d'empêcher une mutation qui briserait ce consensus ; tandis qu'une branche commune aux 16 arbres ne pourra jamais être affectée par une mutation brisant son consensus. Les consensus permettent donc de fixer une partie de la topologie de l'ensemble des arbres (quelque soit leur population), si elle est présente dans une majorité d'entre eux. Il ne faut cependant pas écarter la possibilité qu'un consensus puisse émerger par hasard, sans que la bipartition qu'il représente ne soit présente dans l'arbre optimal. Il y a alors de fortes chances qu'il s'agisse d'un optimum local, et fixer définitivement cette bipartition si elle est présente dans tous les arbres empêcherait l'heuristique de s'en échapper. Nous avons donc introduit le paramètre de tolérance, permettant de tout de même accepter une mutation brisant un consensus, avec une probabilité donnée par l'utilisateur. Un autre problème est ce que doit faire l'opérateur en cas de mutation refusée à cause d'un consensus. Recommencer l'opération sur une autre cible au hasard pourrait prendre énormément de temps, surtout si le nombre de consensus est important (la configuration des consensus pourrait même rendre un opérateur impossible quelque soit sa cible). Deux solutions sont

proposées à l'utilisateur. La première utilise des opérateurs « aveugles » et annule tout simplement la mutation si elle brise un consensus, l'individu ne sera donc pas muté lors de la génération actuelle. La seconde utilise des opérateurs « supervisés » et propose à l'opérateur une liste de cibles valides vis-à-vis des consensus. La méthode de génération de cette liste varie d'un opérateur à l'autre et la liste n'est pas exhaustive, permettant d'être construite assez rapidement, même quand le nombre de consensus est faible. Nous les détaillerons dans la section Opérateurs de mutation ci-dessous.

La phase de sélection est similaire à celle de l'algorithme génétique, avec les mêmes stratégies de sélection disponibles (voir Table 3) et la possibilité de fixer un taux de recombinaison, et est effectuée indépendamment dans chaque population. La recombinaison étant interne à une population, nous avons introduit un principe de recombinaison inter-population, que nous avons appelé « hybridation » pour la différencier de la recombinaison intra-population. Avant chaque phase de mutation, après avoir créé la liste des consensus, l'algorithme détermine si une phase d'hybridation va avoir lieu, selon une probabilité fixée par l'utilisateur. Si une phase d'hybridation a lieu, une population choisie au hasard ne subira aucune mutation lors de cette itération. Au lieu de cela, chaque individu de cette population (excepté le meilleur) sera recombiné avec un individu compatible d'une autre population. Une fois que chaque arbre de cette population aura été recombiné, les autres populations subiront normalement la phase de mutation. Ajuster le taux d'hybridation permet donc d'augmenter l'échange d'informations entre les différentes populations, offrant au metaGA de la recombinaison inter-population en plus de la recombinaison intra-population.

Les différentes populations subissant les phases de mutation et de sélection indépendamment (seule la génération de la liste de consensus se fait en commun), nous avons implémenté cette version du metaGA en suivant une approche programmatique « parallèle ». Chaque traitement de population (mutations et sélection) est géré dans un nouveau thread, et par défaut, ces threads seront lancés séquentiellement. Un paramètre permet cependant d'introduire le nombre de processeurs que l'utilisateur désire affecter au traitement en parallèle du metaGA. Par exemple, s'il désire y affecter 2 processeurs, le metaGA mettra chaque population à traiter dans un « pool » et lancera 2 threads simultanément. Dès qu'un thread aura terminé de traiter sa population, il en piochera une nouvelle dans ce pool. Toute la procédure n'est cependant pas parfaitement parallélisable. En effet, toutes les populations doivent avoir été traitées avant de pouvoir passer à la génération suivante, et le temps de traitement d'une population à l'autre peut fortement varier. Les performances gagnées dépendent également du nombre de populations choisi par l'utilisateur, et du nombre de processeurs affectés : si 5 populations doivent être traitées par 2 processeurs, il y aura très souvent un processeur « idle » pendant que l'autre traite la dernière population. De plus, comme nous l'expliquerons dans la section « Réplicats et conditions d'arrêt - Parallélisation » (page 57), réaliser la copie d'un arbre doit se faire en grande partie de manière asynchrone, et les copies d'arbre sont nombreuses à la fin de la phase de sélection. Le gain en temps n'est donc pas directement proportionnel au nombre de processeurs utilisés, mais on peut tout de même noter un gain total en temps de l'ordre de 40% en affectant 2 processeurs à un nombre pair de population.

CRITÈRES D'ÉVALUATION

MetaPIGA utilise uniquement le critère de maximum de vraisemblance pour évaluer une phylogénie, sous les différents modèles de substitutions nucléotidiques présentés dans l'introduction (JC, K2P, HKY85, TN93 et GTR). L'utilisateur peut donner directement les paramètres de taux relatifs instantanés dans une matrice ou via un rapport de taux transition:transversion. Il a également la possibilité d'introduire une hétérogénéité des taux via une distribution gamma (il choisit le nombre de catégories et la forme de la distribution), ainsi qu'une proportion d'invariants.

Via le même panneau de configuration, l'utilisateur peut déterminer si les paramètres du modèle et/ou les longueurs de branches doivent être optimisés. L'optimisation utilise un algorithme génétique qui n'optimisera que les paramètres sélectionnés par l'utilisateur (longueurs de branches, taux instantanés de substitution, forme de la distribution gamma, proportion de sites invariants, variation de taux parmi les partitions), aux moments qu'il choisit (jamais, ou uniquement à la fin de l'heuristique, ou régulièrement au cours de l'heuristique selon une certaine probabilité, ou après un nombre fixé d'itérations). Cette fonction d'optimisation séparée de l'heuristique permet au programme de se concentrer (a) d'abord sur la recherche de la topologie optimale (en utilisant principalement des opérateurs de mutation topologiques et sporadiquement des opérateurs de mutations des paramètres ou des longueurs de branches), en évitant de sur-optimiser l'arbre, ce qui en plus de prendre un temps CPU énorme à chaque itération, pousserait l'arbre vers un optimum local ; (b) ensuite sur l'optimisation fine des paramètres du modèles et des longueurs de branche une fois la meilleure topologie choisie. Attention, l'essentiel de l'optimisation des paramètres du modèle et de la longueur des branches s'effectue généralement pendant l'heuristique, l'optimisation en fin de recherche constitue donc un « dernier coup d'optimisation » pour finaliser la solution. L'algorithme d'optimisation offre également la possibilité à l'utilisateur d'estimer automatiquement les paramètres de départ du modèle (paramètres de taux relatif instantané, forme de la distribution gamma et proportion d'invariant). MetaPIGA construit alors le Neighbor-Joining Tree et optimise uniquement ces paramètres, qui seront de bonnes approximations pour le démarrage de l'heuristique, quelque soi(en)t le ou les arbre(s) de départ choisi(s).

Le choix du modèle complet (modèle de substitution, hétérogénéité des taux et proportion d'invariant) ne doit pas se faire au hasard, car choisir un modèle trop complexe peut être aussi néfaste que choisir un modèle trop simple. Utiliser un modèle trop simple amène le risque de sous-évaluer certaines solutions, ou pire, de mal les évaluer si les hypothèses du modèle sont fausses ou incomplètes (par exemple ne pas prendre en compte l'hétérogénéité des taux alors qu'il y en a en réalité). Mais utiliser un modèle trop complexe peut mener à un problème classique en statistique appelé « overfitting », exagérant des fluctuations mineures dans les séquences, sans compter que le temps CPU nécessaire à l'évaluation de la vraisemblance est proportionnel à la complexité du modèle. Différentes méthodes ont été proposées pour estimer quel modèle s'adapte le mieux à un jeu de données, et MetaPIGA en implémente 3, directement accessibles depuis le panneau de configuration des critères d'évaluation. Il s'agit du « Likelihood Ratio Test (LRT) », du « Akaike Information Criterion (AIC) » et du « Bayesian

Information Criterion (BIC) », qui avaient été développés pour MODELTEST (Posada & Crandall, 1998).

Le LRT teste différentes combinaisons de modèles avec ou sans hétérogénéité des taux et avec ou sans sites invariants, et détermine si le gain en vraisemblance d'un modèle plus complexe est significatif. Comme les modèles que nous utilisons sont tous imbriqués les uns dans les autres (JC ⊂ K2P ⊂ HKY85 ⊂ TN93 ⊂ GTR), LRT commence par tester chaque modèle par rapport à celui qui lui est juste supérieur, sans hétérogénéité des taux ou de sites invariants. Une fois le modèle de substitution fixé, il le teste avec ou sans hétérogénéité des taux, puis avec ou sans sites invariants. Le ratio est calculé en estimant le Neighbor Joining Tree (après optimisation des éventuels paramètres du modèle) et est basé sur le test statistique $\delta = -2\log\Lambda$ où Λ est

$$\Lambda = \frac{\max[L_0(NullModel|Data)]}{\max[L_1(AlternativeModel|Data)]} \qquad \textbf{Eq.27}$$

Où L_0 est la vraisemblance sous l'hypothèse null (modèle simple) et L_1 est la vraisemblance sous l'hypothèse alternative (modèle plus complexe). Comme les modèles que nous comparons sont toujours imbriqués, la statistique δ est distribuée asymptotiquement comme une χ^2 avec q degrés de liberté, où q est la différence en nombre de paramètres entre les 2 modèles.

L'AIC est un estimateur asymptotiquement non-biaisé de la quantité informative de Kullback-Leibler (Kullback, Leibler, 1951), qui est une mesure de l'information perdue quand un modèle est utilisé pour approximer la réalité. Sélectionner le modèle avec la valeur d'AIC minimale est approximativement équivalent à minimiser la distance Kullback-Leibler entre le vrai modèle et les données estimées. L'AIC pénalise un nombre croissant de paramètres dans le modèle, prenant ainsi en compte non seulement l'adéquation du modèle mais également la variance des paramètres estimés. L'AIC d'un modèle i est calculé comme ceci :

$$\mathrm{AIC_i} = -2\ \ln \mathrm{L_i} + 2\ \mathrm{K_i} \qquad \textbf{Eq.28}$$

Où $\mathrm{K_i}$ est le nombre de paramètres libres dans le $i$ème modèle et $\mathrm{L_i}$ est la valeur de maximum de vraisemblance du Neighbor Joining Tree sous le $i$ème modèle.

Le BIC quand à lui est calculé de la manière suivante :

$$\mathrm{BIC_i} = -2\ \ln \mathrm{L_i} + \mathrm{K_i}\log n \qquad \textbf{Eq.29}$$

Où $\mathrm{K_i}$ est le nombre de paramètres libres dans le $i$ème modèle, $\mathrm{L_i}$ est la valeur de maximum de vraisemblance du Neighbor Joining Tree sous le $i$ème modèle, et n est le nombre de sites dans l'alignement. Le BIC a été développé comme une approximation du log de la vraisemblance marginale d'un modèle, et donc la différence entre 2 estimations BIC peut être une bonne approximation du logarithme népérien du facteur de Bayes (Kass, Wasserman, 1995). Avec la même probabilité à priori de tous les modèles en compétition, choisir le modèle avec le BIC minimal est équivalent à sélectionner le modèle avec les probabilités postérieures maximales. De cette manière, les poids BIC peuvent être vus comme des approximations des probabilités postérieures du modèle (Wasserman, 2000). Le BIC a tendance à sélectionner des

modèles qui sont moins complexes que les facteurs de Bayes, et si n > 8, le BIC sélectionne des modèles plus simples que l'AIC (Forster, Sober, 2004).

GÉNÉRATION D'UN ARBRE DE DÉPART

Quatre choix sont offerts à l'utilisateur pour générer le ou les arbres de départ. Il peut s'agir d'un arbre fourni par l'utilisateur, d'un arbre totalement aléatoire, de l'arbre généré par la méthode du Neighbor Joining (voir section Méthodes d'inférence phylogénétique page 14) ou d'un arbre pseudo-aléatoire construit par ce que nous avons baptisé le « Loose Neighbor Joining ».

Pour la génération par Neighbor Joining, l'utilisateur peut déterminer de quelle manière les distances sont calculées, en sélectionnant le modèle de substitution nucléotidique (parmi JC, K2P, HKY85, TN93 et GTR), si une hétérogénéité des taux doit être modélisée par une distribution gamma, et s'il y a une proportion de sites invariants. Dans le calcul des distances, la proportion d'invariant est utilisée pour ajuster le nombre total de sites afin d'avoir des distances égales au nombre moyen de substitutions pour les sites variables uniquement. La composition de base des sites invariants peut être spécifiée de trois manières différentes : égale (les sites invariants auront la même composition pour les 4 bases), estimée (les sites invariants reflèteront la composition de bases moyenne parmi les séquences) ou constante (les sites invariants reflèteront la composition de bases des sites qui sont constants).

La génération aléatoire d'arbre fonctionne comme suit : elle démarre d'un nœud racine, et y ajoute les 3 nœuds fils. Chaque nœud fils a 50% de chance d'être un taxa (choisi aléatoirement parmi ceux n'ayant pas encore été insérés dans l'arbre), sinon il s'agit d'un nœud interne. La procédure est répétée récursivement pour chaque nœud interne inséré. Si lors de l'ajout d'un nœud fils il s'agit du dernier emplacement disponible dans l'arbre (c'est-à-dire que tous les autres nœuds terminaux sont des taxa) et qu'il reste plus qu'un taxon à insérer, c'est automatiquement un nœud interne qui est choisi pour ce nœud fils. Les longueurs de branches sont tirées aléatoirement dans une distribution exponentielle négative de paramètre $\lambda = 1$, décalée de 0,001.

Après plusieurs tests, nous avons remarqué qu'utiliser l'arbre de Neighbor Joining comme point de départ pour la recherche heuristique n'était pas recommandé. En effet, il s'agit d'un optimum local dans l'espace de recherche et la méta-heuristique aura du mal à s'en échapper. Les arbres totalement aléatoires quand à eux, sont extrêmement mauvais, et nous n'avons pas trouvé de distribution qui puisse approximer de manière réaliste les longueurs de branches. La méta-heuristique pourra donc mettre très longtemps à sortir de cette « vallée » de l'espace de recherche, surtout pour améliorer les longueurs de branches. Nous avons donc introduit la méthode de « Loose Neighbor Joining » pour générer un arbre de départ ni trop mauvais, et ni trop bon ! L'algorithme est basé sur celui du « Neighbor Joining », mais là où normalement la méthode NJ joint 2 nœuds ayant une distance minimale, notre méthode LNJ construit une liste contenant les $(range \times \frac{(NTax \times (NTax-1))}{2})$ distances les plus courtes et y choisit aléatoirement un couple de nœuds. $NTax$ est le nombre de séquences, et $range$ est un paramètre compris

dans l'intervalle $]0,1[$ et fourni par l'utilisateur. Hormis cela, les longueurs de branches sont calculées normalement en utilisant la méthode NJ. Donc si le paramètre $range$ est proche de 0, l'arbre aléatoire sera proche de l'arbre Neighbor Joining ; s'il est proche de 1, l'arbre aura une topologie complètement aléatoire (mais des longueurs de branches calculées avec la méthode NJ). Comme une matrice de distance est également construite, l'utilisateur peut choisir le modèle de substitution, l'hétérogénéité des taux et la proportion d'invariant utilisés.

OPÉRATEURS DE MUTATION

Les opérateurs de mutation sont les outils qui permettront d'explorer l'espace de recherche, en perturbant la structure d'un arbre. Ces opérateurs sont divisés en 2 groupes : ceux qui modifient uniquement la topologie d'un arbre, et ceux qui modifient ses paramètres. La méta-heuristique utilisera uniquement les opérateurs sélectionnés par l'utilisateur, séquentiellement ou aléatoirement. Pour une sélection aléatoire, une fréquence peut être associée à chaque opérateur, pour favoriser l'utilisation de certains d'entre eux par rapport aux autres. Nous avons également implémenté des fréquences dynamiques, c'est-à-dire qui évoluent au cours du temps selon leurs performances. Après un nombre défini d'itérations de la méta-heuristique, la somme des gains en vraisemblance apportés par chaque opérateur est calculée, et les fréquences sont réajustées proportionnellement. L'utilisateur peut définir une fréquence minimale (pour éviter qu'un opérateur ne soit plus du tout utilisé), et sélectionne quels opérateurs doivent avoir une fréquence dynamique (pouvant ainsi définir certains opérateurs qui auront une fréquence constante). La Table 4 présente un petit exemple pratique.

Table 4 – Exemple de fréquences dynamiques. Seuls les opérateurs NNI et SPR ont une fréquence dynamique et seront ajustés, BLM a une fréquence constante pour toute l'heuristique.

Opérateur	Fréquence	Dynamique
NNI	40%	Oui
SPR	40%	Oui
BLM	20%	Non

Gain en vraisemblance cumulé après 100 itérations	
NNI	1500
SPR	1000
BLM	580

Fréquences ajustées pour les 100 prochaines itérations	
NNI	48%
SPR	32%
BLM	20%

NNI (NEAREST NEIGHBOR INTERCHANGE)

Le NNI sélectionne une branche interne comme candidat, puis échange l'une des 2 branches filles de droite avec l'une des 2 branches filles de gauche. En nous basant sur la Figure 13 au-dessous, le NNI a sélectionné la branche rouge comme candidat et pourra échanger le sous-arbre {C,D} ou {A,B,E,F} avec le sous-arbre {I} ou {H,G}. Les deux seuls résultats possibles sont montrés sous les flèches vertes.

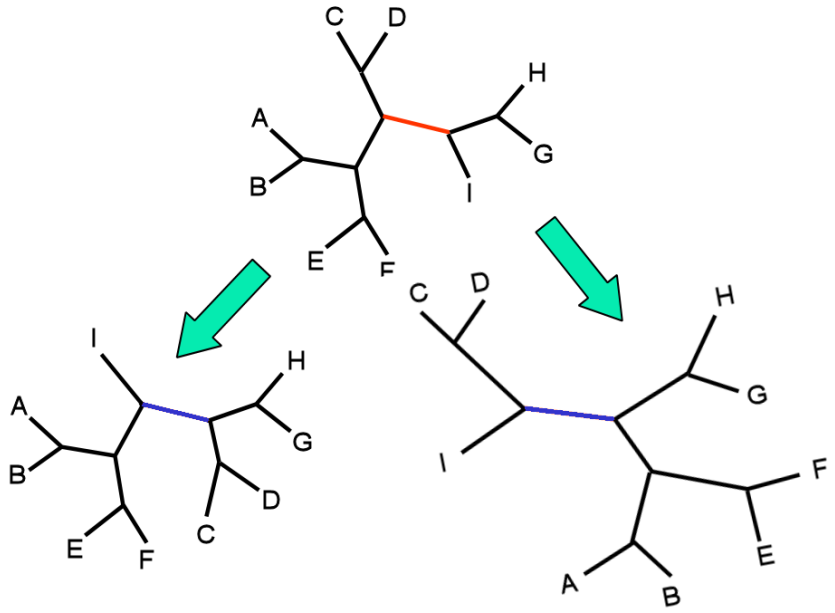

Figure 13 – NNI (Nearest Neighbor Interchange)

Si un outgroup de taille > 2 est défini, la branche menant de la racine vers l'outgroup n'est jamais sélectionnée comme candidat, pour éviter de briser l'outgroup. La restriction imposée au NNI par un consensus est assez simple à gérer, car en considérant la bipartition générée par la branche candidate, NNI fait systématiquement passer un groupe de taxa d'une partition à l'autre. Il suffit donc de vérifier que la branche choisie comme candidat ne fasse pas partie de la liste de consensus. La version supervisée du NNI choisit automatiquement son candidat parmi les branches internes ne se trouvant pas dans la liste des consensus.

SPR (SUBTREE PRUNING AND REGRAFTING)

SPR sélectionne n'importe quelle branche comme candidat, la « coupe » et la « ré-attache » à un autre endroit de l'arbre. Dans l'exemple présenté sur la Figure 14, la branche rouge est sélectionnée comme candidat, et coupée, c'est-à-dire que le nœud interne (appelons-le X) ayant comme descendants les sous-arbres {C,D} et {A,B,E,F} a été retiré, et que {C,D} est directement relié à {A,B,E,F} par une nouvelle branche unique. Ensuite, une branche cible est sélectionnée au hasard, la branche menant au taxa A dans notre exemple. Le nœud X est réinséré entre {A} et {B,C,D,E,F}, créant 2 nouvelles branches menant l'une au sous-arbre {A} et

l'autre au sous-arbre {B,C,D,E,F}. Notez que SPR ne ré-attache jamais une branche à l'endroit où elle a été détachée, modifiant donc systématiquement la topologie de l'arbre.

En cas de consensus, une liste des branches cibles valides est systématiquement construite. Il s'agit d'une procédure récursive, qui démarre de la branche « coupée », ajoutant à la liste des cibles valides les branches filles qui ne font pas partie de la liste de consensus. Pour chaque branche valide qui n'est pas une branche terminale, la procédure est répétée. SPR choisit alors au hasard une branche valide dans cette liste. La liste pouvant bien entendu être vide selon le candidat choisi, le comportement du SPR varie selon qu'il soit supervisé ou non. En cas de SPR aveugle, le SPR s'arrête sans faire de mutation s'il n'y a pas de cible valide pour son candidat. En cas de SPR supervisé, le candidat est éliminé et un nouveau candidat est choisi parmi les branches restantes, et une nouvelle liste des cibles valides est générée. Cette procédure est répétée tant que le SPR n'a pas sélectionné de candidat ayant au moins une cible valide, ou qu'il arrive à court de candidats, auquel cas SPR s'arrête sans effectuer de mutation (nous nous trouvons alors dans un cas où les consensus empêchent toute mutation par SPR). Si un outgroup de taille > 2 est défini, SPR vérifie également qu'un sous-arbre de l'outgroup ne soit pas ré-attaché dans l'ingroup, et inversement. En cas de conflit, une autre cible est choisie parmi les cibles valides, ou le SPR est annulé s'il n'y en a plus de disponible.

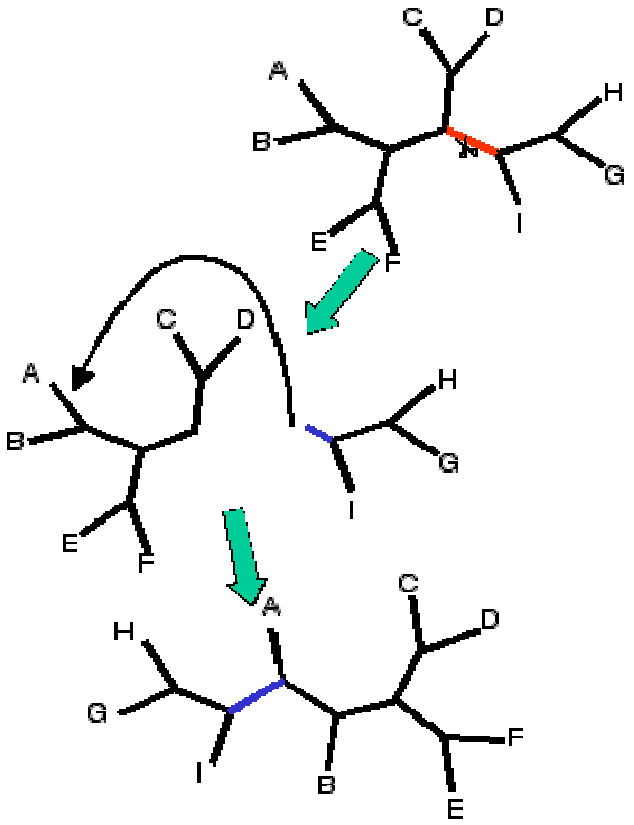

Figure 14 – SPR (Subtree Pruning and Regrafting)

TBR (TREE BISECTION RECONNECTION)

TBR sélectionne n'importe quelle branche comme candidat, et la brise (retirant de l'arbre les 2 nœuds qui la délimitent), séparant donc l'arbre en 2 sous-arbres. Ces 2 sous-arbres sont ensuite reconnectés en sélectionnant au hasard une branche cible dans chacun d'eux. Dans l'exemple présenté sur la Figure 15, la branche rouge a été sélectionnée comme candidat. Elle est retirée de l'arbre, séparant l'arbre en 2 sous-arbres : {A,B,C,D,E,F} et {G,H,I}. Les 2 nœuds de la branche candidats sont retirés avec elle, fusionnant en une seule branche les 2 branches filles qu'ils avaient dans chaque sous-arbre. Ensuite une branche cible est choisie au hasard dans le premier sous-arbre (ici la branche menant au taxa A). Une branche cible doit également être choisie dans le deuxième sous-arbre, et la Figure 15 nous montre les trois solutions possibles selon la branche choisie. Notez que les branches cibles sont systématiquement différentes des branches où était attachée la branche candidate, TBR modifiant donc systématiquement la topologie.

En cas de consensus, la procédure permettant de construire les cibles valides du SPR est utilisée dans chaque sous-arbre. Hormis le fait que 2 listes sont utilisées, le principe est exactement le même en TBR qu'avec la version correspondante de SPR (aveugle / supervisée). La même vérification pour l'outgroup est faite comme en SPR également.

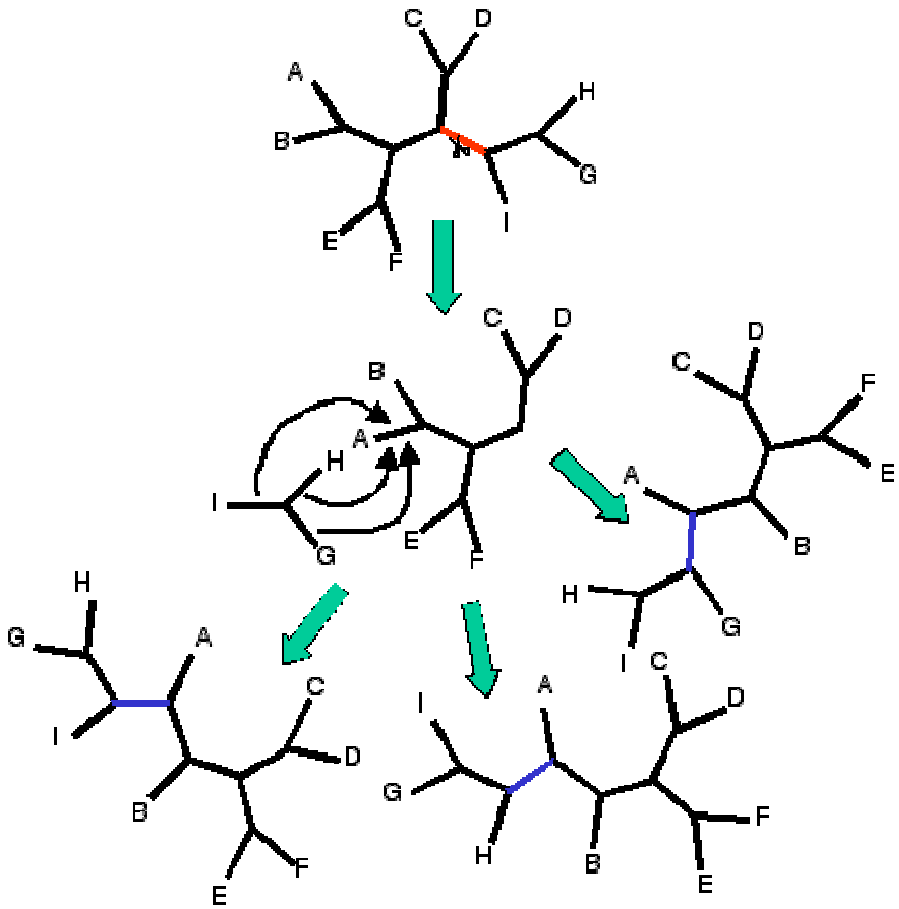

Figure 15 – TBR (Tree Bisection and Reconnection)

TXS (TAXA SWAP)

Un paramètre n est associé à TXS, et détermine le nombre de nœuds choisis comme candidats. TXS choisit n taxa (des nœuds externes donc) au hasard, et les permute entre eux aléatoirement. Si $n = 2$, les 2 taxa échangent systématiquement leurs positions, mais pour $n > 2$ il se peut, dans de rares cas, que la permutation soit semblable à l'état initial. Le paramètre n peut également être mis à la valeur « random », TXS permutant alors un nombre aléatoire de taxa à chaque utilisation. Si un outgroup de taille > 2 est défini, TXS aura une chance proportionnelle à la taille de l'outgroup de ne sélectionner que des taxa appartenant à l'outgroup. Sinon il choisira systématiquement des taxa appartenant à l'ingroup. En cas de consensus, TXS vérifie pour chaque bipartition de la liste de consensus que tous les taxa à permuter se trouvent bien sur la même partition. En effet, permuter un taxa d'une partition vers une autre briserait ce consensus. Pour le TXS supervisé, un ensemble valide de taxas appartenant à une même partition du consensus est généré, afin d'être sûr que le TXS ne violera aucun consensus. Cet ensemble est choisi parmi toutes les partitions de la liste de consensus qui sont plus grandes où égales à n. Le choix est aléatoire, mais chaque partition a une chance d'être choisie proportionnelle à sa cardinalité.

STS (SUBTREE SWAP)

STS est une généralisation de TXS aux nœuds internes, ayant donc pour effet de permuter des sous-arbres (pouvant très bien ne contenir qu'un seul taxa). Un paramètre n est associé à STS, mais il ne peut valoir que 2 ou « random ». En effet, dans TXS le paramètre n est simplement limité au nombre de taxa. Dans STS par contre, sa limite évolue à chaque nœud interne sélectionné pour être permuté. En effet, un nœud interne ne peut pas être permuté avec l'un de ses descendants. Permuter uniquement 2 nœuds internes est donc toujours possible. La valeur « random » est quand à elle un peu particulière, STS divisant l'arbre complet en un nombre aléatoire de sous-arbres qui seront permutés. STS sélectionne toujours des nœuds internes appartenant à l'ingroup, pour éviter toute violation de l'outgroup. En cas de consensus, STS ne peut pas briser de consensus provenant d'une branche interne à l'un des sous-arbres à permuter, ni d'une branche mère d'un sous-arbre à permuter. Il ne vérifie donc que les autres branches de la liste de consensus, en s'assurant que tous les taxa d'un même sous-arbre à permuter se trouvent bien sur la même partition. STS n'a pas de version supervisée, et annule toujours une mutation qui viole un consensus, générer une liste de nœuds interne valide étant bien trop gourmand en ressources.

BLM ET BLMINT (BRANCH LENGTH MUTATION)

BLM sélectionne une branche de l'arbre au hasard et multiplie sa longueur par un facteur aléatoire, tiré dans une distribution exponentielle de paramètre $\lambda = 2$, décalée de 0,5 afin d'avoir un minimum de 0,5 et une moyenne de 1. Lors d'une mutation, une longueur de branche peut donc être réduite au maximum d'un facteur 2, et sera rarement allongée d'un plus grand facteur. BLMint fait exactement la même chose, mais cible toujours une branche interne. Il peut être utilisé si on considère que la longueur des branches externes est plus stable et doit donc être moins souvent mutée. Comme il ne s'agit pas d'un opérateur topologique, il n'est aucunement affecté par un consensus ou un éventuel outgroup.

RPM (RATE PARAMETERS MUTATION)

RPM s'accompagne d'un paramètre pouvant valoir 1 ou le nombre de paramètres total associé au modèle de substitution nucléotidique choisi. Il choisi l'un des paramètres du modèle de substitution au hasard (ou tous selon le paramètre choisi), et le multiplie par un facteur aléatoire, tiré dans une distribution exponentielle de paramètre $\lambda = 2$, décalée de 0,5 afin d'avoir un minimum de 0,5 et une moyenne de 1. Si le jeu de données est partitionné à l'aide de charsets, le ou les paramètres ne sont mutés que pour un seul charset, sélectionné au hasard. Comme il ne s'agit pas d'un opérateur topologique, il n'est aucunement affecté par un consensus ou un éventuel outgroup.

GDM (GAMMA DISTRIBUTION MUTATION)

GDM multiplie le paramètre de forme de la distribution gamma (en cas d'hétérogénéité des taux) par un facteur aléatoire, tiré dans une distribution exponentielle de paramètre $\lambda = 2$, décalée de 0,5 afin d'avoir un minimum de 0,5 et une moyenne de 1. Si le jeu de données est partitionné à l'aide de charsets, le paramètre n'est muté que pour un seul charset, sélectionné au hasard. Comme il ne s'agit pas d'un opérateur topologique, il n'est aucunement affecté par un consensus ou un éventuel outgroup.

PIM (PROPORTION OF INVARIANT MUTATION)

PIM multiplie la proportion de sites invariants par un facteur aléatoire, tiré dans une distribution normale, mais devant être supérieur à 0,4. Si le jeu de données est partitionné à l'aide de charsets la proportion de sites invariants n'est mutée que pour un seul charset, sélectionné au hasard. Comme il ne s'agit pas d'un opérateur topologique, il n'est aucunement affecté par un consensus ou un éventuel outgroup.

APRM (AMONG-PARTITION RATE MUTATION)

APRM ne peut être utilisé que si le jeu de données a été partitionné à l'aide de charsets, et va muter les paramètres de variation de taux parmi ces partitions. Étant donné que les taux relatif Θ_p sont contraints d'avoir une moyenne pondérée égale à 1 (voir Eq.25), deux charsets (1 et 2) vont être choisis au hasard, et leurs taux relatifs associés mutés en concordance. Θ_1 va être multiplié par un facteur aléatoire, tiré dans une distribution normale, mais devant être supérieur à 0,4. Θ_2 va ensuite être ajusté pour que l'Eq.25 reste valide. Comme il ne s'agit pas d'un opérateur topologique, il n'est aucunement affecté par un consensus ou un éventuel outgroup.

RÉPLICATS ET CONDITIONS D'ARRÊT

MetaPIGA peut être configuré pour lancer séquentiellement plusieurs analyses avec exactement les mêmes paramètres de départs (des réplicats). Tous les arbres solutions générés par ces réplicats (l'échantillonnage) seront utilisés pour générer un arbre consensus avec des « valeurs de support MetaPIGA » pour chaque branche (Lemmon & Milinkovitch 2002). Par exemple, si l'heuristique choisie est le metaGA avec 4 populations de 4 individus, et que l'utilisateur fixe le nombre de réplicats à 250, MetaPIGA lancera automatiquement 250 analyses, générant 1000 arbres solutions tous utilisés pour construire un arbre consensus. L'arbre consensus est généré par « majority-rule », c'est-à-dire que les branches communes dans plus 50% des arbres font automatiquement partie du consensus (les partitions qu'elles génèrent sont par définition toutes compatibles entre elles). Les autres branches formant des partitions incompatibles avec celles-ci sont éliminées, et les branches restantes sont sélectionnées une par une, par ordre décroissant de représentation dans l'échantillonnage, en éliminant à chaque ajout d'une nouvelle branche celles qui sont incompatibles avec celle-ci. Le pourcentage de représentation dans l'échantillonnage est utilisé comme valeur de support de branche. Les longueurs de branches et les paramètres du modèle utilisés pour cet arbre consensus sont une moyenne des valeurs des échantillons sélectionnés, permettant de calculer la vraisemblance de l'arbre de consensus.

Il est cependant difficile d'estimer le nombre de réplicats nécessaires pour obtenir un arbre consensus avec un support statistique suffisant. Pour éviter de devoir générer plus de réplicats que nécessaire, étant donné que la génération d'un seul réplicat peut être très longue, nous avons implémenté une méthode qui permet de stopper la génération de réplicats quand l'erreur moyenne relative (« Mean Relative Error » = MRE) entre une série d'arbres consensus reste en dessous d'un certain seuil. L'erreur moyenne relative entre 2 arbres consensus T_i et T_j est calculée comme suit :

$$MRE(T_i, T_j) = \sum_{p}^{nPartition} \left| \frac{\Phi_{T_i}^{p} - \Phi_{T_j}^{p}}{\max\left(\Phi_{T_i}^{p} \middle| \Phi_{T_j}^{p}\right)} \right| \Big/ nBranch \qquad \text{Eq.30}$$

Où $nPartition$ est égal au nombre de partitions différentes dans l'ensemble des 2 arbres consensus T_i et T_j, $nbranch$ est le nombre de branches d'un arbre consensus et $\Phi_{T_i}^p$ est la valeur de support de la partition p dans l'arbre T_i, et sachant que :

$$\left|\frac{\Phi_{T_i}^p - \Phi_{T_j}^p}{\max\left(\Phi_{T_i}^p \middle| \Phi_{T_j}^p\right)}\right| = 1 \; si \begin{cases} \Phi_{T_i}^p \text{et } \Phi_{T_j}^p \text{sont égaux à } 0 \\ \Phi_{T_i}^p \text{ ou } \Phi_{T_j}^p \text{ n'existe pas} \end{cases} \quad \text{Eq.31}$$

L'erreur moyenne relative entre deux arbres consensus nous permet donc de quantifier l'apport du dernier réplicat aux supports de branches : plus la MRE est proche de zéro, moins l'apport est significatif. Une branche qui avait une faible valeur de support dans T_i qui s'est améliorée dans T_j aura un poids important dans la valeur de MRE, tandis qu'une branche déjà très bien supportée dans T_i affectera peu la valeur de MRE même si elle s'est améliorée dans T_j. De plus, l'apport à la valeur de MRE est toujours proportionnel au nombre de branches. La table ci-dessous montre quelques exemples avec l'apport relatif à la valeur de MRE.

Table 5 – Quelques exemples d'apport à l'erreur moyenne relative entre deux arbres consensus.

Support T_i	Support T_j	Apport MRE	
94%	96%	2%	Bonne valeur de support, différence mineure, apport très mineur à la MRE.
94%	100%	6%	Bonne valeur de support, différence importante, apport mineur à la MRE.
30%	32%	6%	Mauvaise valeur de support, différence mineure, apport mineur à la MRE.
30%	36%	17%	Mauvaise valeur de support, différence importante, apport important à la MRE.
80%	100%	20%	Bonne valeur de support, différence très importante, apport important à la MRE.
30%	50%	40%	Mauvaise valeur de support, différence très importante, apport très important à la MRE.
100%	inexistant	100%	Si une branche apparaît ou disparaît, l'apport à la MRE est maximal.
0%	0%	100%	Si les 2 branches ont une valeur de support nulle, l'apport à la MRE est maximal.

Grâce à cette erreur moyenne relative nous pouvons déterminer qu'un arbre consensus se stabilise si sa valeur de MRE reste en dessous d'un certain seuil parmi un certain nombre d'arbres consensus consécutifs. Afin d'éviter de nous arrêter trop tôt, nous allons tout de même produire un nombre minimal de réplicats, par sécurité. Par exemple imaginons que nous voulions que l'arbre de consensus ait une valeur de MRE inférieure à 5% parmi au moins 10 échantillons consécutifs, et que nous voulons générer au minimum 100 réplicats. Dans ce cas, nous allons générer 100 réplicats et sauvegarder l'arbre consensus obtenu (T_{100}), puis calculer son MRE avec l'arbre consensus obtenu après 101 réplicats (T_{101}). Si $MRE(T_{100}, T_{101})$ est inférieure à 5%, nous calculerons $MRE(T_{100}, T_{102})$ et ainsi de suite tant que nous restons

sous les 5%. Si nous arrivons à calculer $MRE(T_{100}, T_{110})$ et qu'il est toujours inférieur à 5%, nous stoppons la génération de réplicats. Par contre, dès qu'un MRE passe au-delà de 5%, nous sauvegardons cet arbre consensus comme nouvelle référence de comparaison. Si dans notre exemple $MRE(T_{100}, T_{101})$, $MRE(T_{100}, T_{102})$ et $MRE(T_{100}, T_{103})$ sont tous les 3 inférieurs à 5% mais que $MRE(T_{100}, T_{104})$ ne l'est plus, nous continuons à générer des réplicats et nous réinitialiseront le compteur à zéro. Il faudra à nouveau 10 MRE consécutives inférieures à 5% avec à présent T_{104} en référence, le prochain test étant $MRE(T_{104}, T_{105})$ puis, s'il est inférieur à 5%, $MRE(T_{104}, T_{106})$, et ainsi de suite. L'utilisateur peut paramétrer le nombre minimal et maximal de réplicats à générer, le seuil de MRE et le nombre d'arbres consensus consécutifs devant rester sous ce seuil pour pouvoir stopper la génération.

CONDITIONS D'ARRÊT DES MÉTA-HEURISTIQUES

Nous pouvons également utiliser la MRE comme condition d'arrêt pour le metaGA. En effet, le metaGA disposant de plusieurs populations d'individus à chaque génération, nous avons suffisamment d'arbres pour générer un arbre consensus, et comparer ses valeurs de support à celui d'une génération ultérieure. Nous n'allons cependant pas les comparer à chaque génération, les arbres de consensus ayant tendance à peu varier entre 2 générations consécutives (tout un ensemble de mutations ayant pu être rejeté via une forte pression sélective par exemple). Il vaut donc mieux échantillonner toutes les X générations. Le principe est donc le même que la condition d'arrêt des réplicats, avec cette différence : le metaGA s'arrête lorsque l'erreur relative moyenne parmi un nombre donné d'arbres de consensus reste en dessous d'un seuil donné. Chaque arbre de consensus est construit en utilisant tous les arbres de toutes les populations d'une génération, en échantillonnant après un nombre donné de générations. En plus de cette condition d'arrêt particulière implémentée spécifiquement pour le metaGA (ou toute future heuristique générant de nombreux arbres par itération), nous avons implémenté 3 conditions d'arrêt standards, communes à toutes les heuristiques. L'utilisateur peut choisir un nombre fixe d'itérations, un temps d'exécution maximal ou un nombre d'itérations consécutives pendant lesquelles la vraisemblance du meilleur arbre ne s'améliore pas de plus de 0,01%. Le choix d'une condition d'arrêt n'est pas exclusif, l'utilisateur pouvant en choisir plusieurs, auquel cas l'heuristique s'arrêtera dès qu'au moins une des conditions choisies est rencontrée.

PARALLÉLISATION

La génération de chaque réplicat est indépendante des autres, et l'ordre dans lequel ils sont générés n'a aucune importance. C'est donc une tâche qui se prête bien à la parallélisation. Grâce au multithreading de Java, nous proposons à l'utilisateur de générer plusieurs réplicats en parallèle, il lui suffit pour cela d'indiquer combien d'analyses il désire faire en même temps. Générer 2 réplicats en parallèle avec au moins 2 processeurs à disposition permet en général de réduire le temps total nécessaire de plus de 40%. Ces 2 analyses semblant totalement indépendantes, nous nous sommes demandés pourquoi le gain en temps n'était pas plus proche des 50% que l'on aurait pu escompter. Après divers tests et « profiling » d'analyses, il semblerait que le problème soit dû à l'initialisation et la copie de grandes « arrays », auquelles MetaPIGA fait intensément appel. Nous avons réalisé un test, visant à initialiser et copier 1.000 fois une array de 500.000 réels « double précision ». En réalisant ce test 2x de suite (en série donc), chaque test à eu un temps d'exécution de +/- 3.600 millisecondes pour un total de +/- 7.200 millisecondes. Ensuite, nous avons réalisé les mêmes 2 tests en parallèle (sur une machine ayant un processeur à 4 cœurs). Chaque test a eu un temps d'exécution de +/- 6.500 millisecondes, pour un total de +/- 6.500 millisecondes. Le gain en temps est donc très faible, chaque test prenant presque le double du temps d'exécution. Il semblerait donc que l'initialisation et la copie d'array soit une tâche très peu parallélisable sur une JVM. Toutes les heuristiques de MetaPIGA nécessitent de copier au moins un arbre par itération (pour sauvegarder l'état précédent une mutation). Et copier un arbre nécessite de copier les informations relatives au calcul de la vraisemblance (vu leur important temps de calcul), qui sont stockées dans $4 \times (nTax \times 2 - 1)$ arrays d'une taille égale à la longueur des séquences compressées. Ceci explique donc le gain en temps inférieur à 50% pour nos réplicats générés en parallèle, mais il reste tout de même considérable par rapport à une génération séquentielle. De plus, il dépend de la complexité du modèle choisi et de la taille du jeu de données. En effet, avec un grand jeu de données et un modèle très complexe comme GTR avec hétérogénéité des taux, le processeur est beaucoup plus sollicité que pour un petit jeu de données utilisant le modèle JC. Le gain en temps en parallélisant sera donc plus proche des 50%, car le processeur ne passera pas la majorité de son temps à copier des arrays. Enfin, notons que le gain en temps est proportionnel au nombre de processeurs disponibles, par exemple générer 4 réplicats en parallèle avec 4 processeurs à disposition permet de diminuer le temps d'exécution total de 60 à 70% selon les données et les paramètres choisis.

OUTILS D'ANALYSE

La Figure 16 ci-dessous est une capture d'écran de la fenêtre de progression d'une analyse. En plus des informations textuelles et de barres de progressions (itérations effectuées, temps restant, progression du MRE, nombre de réplicats), on y trouve un graphe montrant la progression de la vraisemblance de la meilleure solution trouvée. Avec le Simulated Annealing, un graphe montrant la courbe de température est également affiché, et avec le metaGA on peut voir la progression simultanée des différentes populations. Si un seul réplicat est généré, on peut également afficher l'arbre de la meilleure solution actuelle ; et en cas de réplicats multiples l'arbre consensus construit avec les réplicats déjà produits. Un bouton stop permet de stopper l'analyse à tout moment, MetaPIGA terminant l'itération en cours avant de s'arrêter comme si une condition d'arrêt avec été rencontrée.

Figure 16 – Une analyse par metaGA avec plusieurs réplicats. La partie supérieure gauche est un « log » reprenant les paramètres de l'analyse, les principales étapes et les éventuelles erreurs. La partie inférieure gauche affiche un graphe reprenant l'évolution de vraisemblance des meilleures solutions de chaque population (la ligne rouge représente la vraisemblance du meilleur arbre de départ). Une ligne cyan indique qu'un nouveau réplicat est commencé. La partie droite affiche l'arbre consensus construit avec les réplicats déjà terminé, et on peut suivre une progression de la valeur de MRE juste en dessous.

Il est également possible de lancer MetaPIGA en ligne de commande, avec un fichier NEXUS contenant les paramètres de l'analyse. Dans ce cas, de simples barres de progression « textuelles » et la valeur de vraisemblance du meilleur arbre trouvé sont affichées.

Nous avons aussi pourvu MetaPIGA d'un mode de lancement d'analyse en « batch ». Ce mode permet à l'utilisateur de facilement configurer toute une série d'analyses et de les lancer séquentiellement sans son intervention. Lancer une série d'analyse impliquant souvent de tester un même jeu de données avec des paramètres différents, ou à l'inverse de tester un même ensemble de paramètres sur une série de jeu de données, l'interface graphique permet de facilement effectuer ces opérations. L'utilisateur peut ouvrir plusieurs fichiers NEXUS dans MetaPIGA, et ouvrir plusieurs fois le même fichier est permis (MetaPIGA assignera à chacun un identifiant unique pour les différencier). L'utilisateur peut également dupliquer un jeu de données ouvert (avec l'ensemble de ses paramètres), ou appliquer les paramètres d'un jeu de données sélectionné à un autre jeu de données ouvert. L'utilisateur n'a plus qu'à ordonner la liste des jeux de données ouverts dans l'ordre de son choix, puis de lancer une analyse « batch ». Une fenêtre d'analyse différente de celle présentée est utilisée, affichant les données minimales de progression de chaque analyse, mais affichant une progression générale du batch et permettant de stopper une analyse pour passer à la suivante, ou de stopper l'entièreté du batch. Enfin, l'utilisateur a la possibilité de sauver son « batch » d'analyses dans un fichier NEXUS. Ce fichier contiendra autant de blocs METAPIGA (paramètres) et DATA (jeu de données) que de jeu de données utilisés dans le batch, et un bloc BATCH reprenant les identifiants de chaque jeu de données et la configuration générale du batch. Il est donc possible de sauvegarder la configuration d'un batch tout entier, ou de le lancer en ligne de commande après l'avoir préparé à l'aide de l'interface graphique.

OUTILS PÉRIPHÉRIQUES INTÉGRÉS

Nous avons intégré à MetaPIGA une série d'outils qui nécessitent habituellement d'utiliser d'autres logiciels, et qui fournissent ici de précieuses fonctionnalités facilement accessibles.

Le TreeViewer (Figure 17) est un outil de visualisation d'arbres. Il affiche les longueurs de branches et les valeurs de support des branches, permet de re-raciner l'arbre sur un autre nœud interne, d'exporter l'arbre en format Newick, de l'imprimer, mais surtout de calculer la vraisemblance de l'arbre affiché. Il est également possible de changer le modèle utilisé et ses paramètres pour chaque charset (ou d'optimiser directement ces paramètres), et de réévaluer la vraisemblance. Les arbres solutions générés par une analyse peuvent directement être transférés au TreeViewer, mais l'utilisateur peut également y charger des arbres contenus dans un fichier NEXUS, ou directement encoder un arbre au format Newick dans la zone de texte en bas à gauche de la fenêtre (très utile pour copier-coller les arbres sauvegardé dans les fichiers de sortie (« logs ») générés par MetaPIGA).

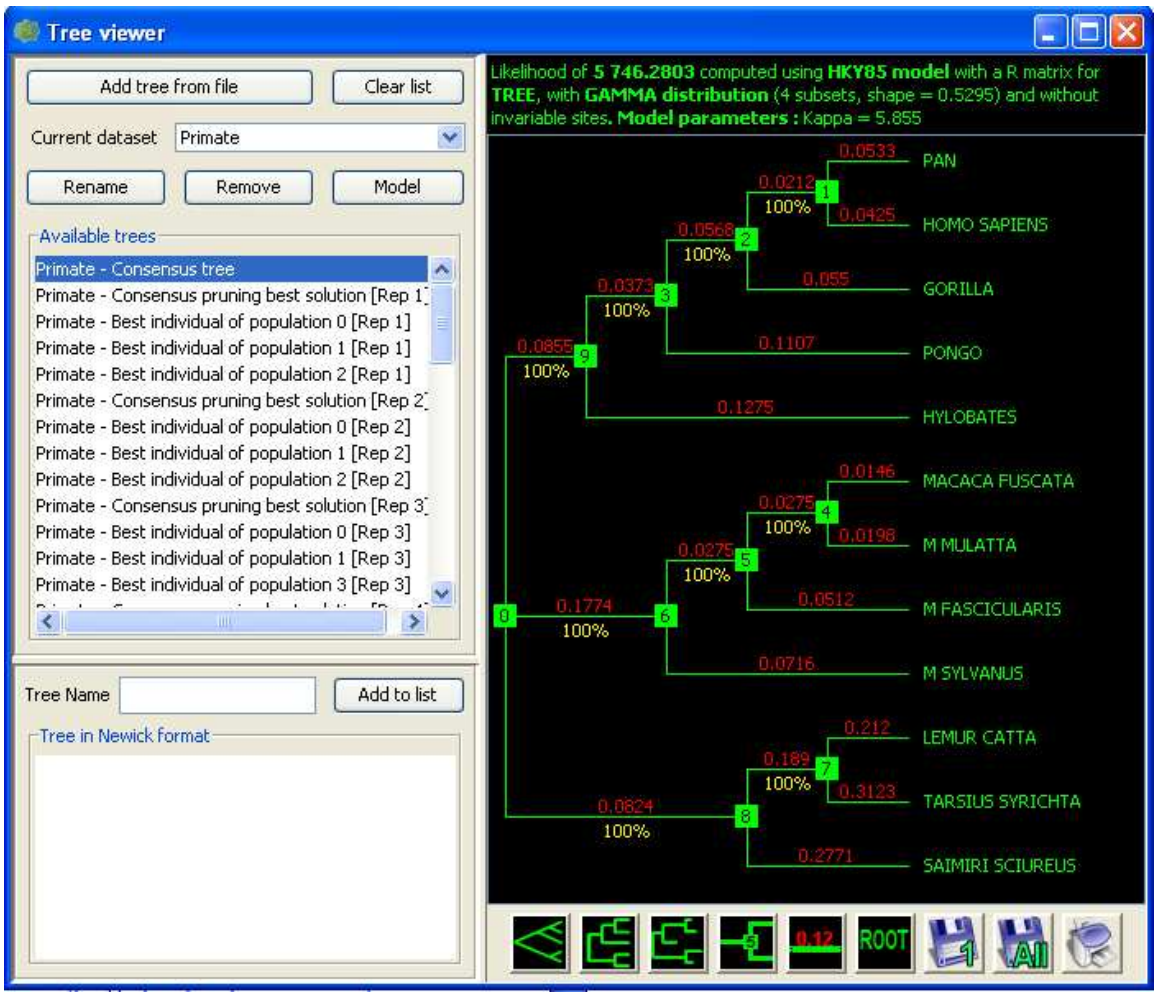

Figure 17 – Le TreeViewer.

Un générateur d'arbre est également implémenté. Il propose les mêmes options que celles disponibles pour générer l'arbre de départ d'une heuristique, et permet d'entrer le nombre d'arbres désirés. Il peut donc servir par exemple pour facilement générer l'arbre de Neighbor Joining ou une série d'arbres aléatoires. Bien sûr, les arbres générés peuvent être directement transférés au TreeViewer.

Un autre outil permet quand à lui de générer immédiatement un arbre de consensus (en suivant le même algorithme de « majority-rule » que celui utilisé pour les consensus de réplicats). Les arbres à utiliser comme échantillonnage sont sélectionnés par l'utilisateur via le TreeViewer, et l'arbre consensus peut bien entendu y être transféré.

OPTIMISATION DU CALCUL DE LA VRAISEMBLANCE

Le calcul de la vraisemblance d'un arbre est une opération critique au niveau des performances de MetaPIGA. Le cœur du calcul est une série d'opérations simples dans le meilleur des cas (pour les modèles bénéficiant d'une équation analytique permettant de calculer les probabilités de transition), mais nécessitant de calculer les valeurs propres, vecteurs propres et l'inverse d'une matrice pour les modèles les plus complexes. Pour calculer la vraisemblance conditionnelle d'un seul nœud de l'arbre, ces opérations sont répétées autant de fois qu'il y a de sites, la totalité étant multipliée par le nombre de nœuds internes de l'arbre. Le temps de calcul total croît donc de manière linéaire avec la longueur des séquences et de manière exponentielle avec le nombre de séquences. Nous avons fait plusieurs ajustements au calcul de la vraisemblance pour améliorer ses performances générales.

Tout d'abord, la vraisemblance de chaque site est indépendante des autres sites, ou de l'ordre dans lequel ils sont estimés. Deux sites formés du même alignement de nucléotides auront donc exactement la même vraisemblance. Pour éviter de faire plusieurs fois une même série de calculs menant au même résultat, nous allons compresser le jeu de données. C'est-à-dire que lorsque n sites de l'alignement de séquences seront égaux, nous n'en garderons qu'un seul et lui donnerons un poids égal à n. Cette technique nous permet de réduire sensiblement la longueur de l'alignement, mais dépend bien entendu du degré de similarité des données.

Ensuite, nous allons tenter de recalculer le moins possible la vraisemblance d'un arbre, ou uniquement les parties de l'arbre ayant changées. Si nous effectuons une quelconque mutation dans un arbre, sa vraisemblance à la racine va évidemment changer, mais les vraisemblances conditionnelles de certains nœuds seront toujours les mêmes. La détermination des nœuds à recalculer dépend de l'opérateur de mutation utilisé, mais ils utilisent tous un système de marquage, et seule la vraisemblance conditionnelle des nœuds marqués sera recalculée. La table 6 décrit quels nœuds seront marqués selon l'opérateur, sachant que lorsque MetaPIGA marque un nœud, tous les ancêtres menant de ce nœud à la racine sont également marqués.

Table 6 – Marquage des nœuds à réévaluer selon l'opérateur

Opérateur	Nœud devant être recalculés
NNI	Les 2 nœuds formant la branche sélectionnée comme candidat, et leurs voisins directs.
SPR, TBR	Tous les nœuds de l'arbre sont marqués, il peut être très compliqué de savoir quels nœuds doivent être recalculés selon l'endroit où se trouve la racine.
TXS, STS	Chaque nœud qui a été sélectionné pour être permuté.
BLM, BLMint	Les 2 nœuds formant la branche dont on a muté la longueur.
RPM, GDM, PIM, APRM	L'arbre entier est réévalué, les paramètres mutés entrant dans le calcul de vraisemblance de chaque nœud.

METAPIGA 2.0 : MAXIMUM LIKELIHOOD LARGE PHYLOGENY ESTIMATION USING THE METAPOPULATION GENETIC ALGORITHM AND OTHER STOCHASTIC HEURISTICS

L'article « MetaPIGA 2.0: maximum likelihood large phylogeny estimation using the metapopulation genetic algorithm and other stochastic heuristics » a été soumis à BMC Bioinformatics en février 2010. Il sera disponible sur le site http://www.metapiga.org.

TITLE AND ABSTRACT

MetaPIGA v2.0: maximum likelihood large phylogeny estimation using the metapopulation genetic algorithm and other stochastic heuristics

Raphaël Helaers[1] and Michel C. Milinkovitch[2]

(1) *Department of Biology, Facultés Universitaires Notre-Dame de la Paix (FUNDP), rue de Bruxelles 61, 5000 Namur, Belgium*; raphael.helaers@fundp.ac.be
(2) *Laboratory of Artificial & Natural Evolution (LANE), Dept. of Genetics & Evolution Sciences III, 30, Quai Ernest-Ansermet, 1211 Genève 4, Switzerland;* Michel.Milinkovitch@unige.ch

Corresponding author:

Michel C. Milinkovitch

Laboratory of Artificial & Natural Evolution (LANE), Dept. of Genetics & Evolution

Sciences III, 30, Quai Ernest-Ansermet, 1211 Genève 4, Switzerland.

Tel +41(0)22 379 67 85

Fax +41(0)22 379 67 95

URL: www.lanevol.org

E-mail: Michel.Milinkovitch@unige.ch

Running Title: MetaPIGA2

Keywords: Phylogeny Inference, Maximum Likelihood, Genetic Algorithm, Stochastic Heurstics, Optimization.

Background. The development, in the last decade, of stochastic heuristics implemented in robust application softwares have made large phylogeny inference a key step in most comparative studies involving molecular sequences. Still, the choice of a phylogeny inference software is often dictated by a combination of parameters not related to the raw performance of the implemented algorithm(s) but rather by practical issues such as ergonomics and/or the availability of specific functionalities.

Results. Here, we present MetaPIGA2, a robust implementation of several stochastic heuristics for large phylogeny inference (under maximum likelihood), including a Simulated Annealing algorithm, a classical Genetic Algorithm, and the Metapopulation Genetic Algorithm (metaGA) together with complex substitution models, discrete Gamma rate heterogeneity, and the possibility to partition data. MetaPIGA2 also implements the Likelihood Ratio Test, the Akaike Information Criterion, and the Bayesian Information Criterion for selecting substitution models that best fit the data. MetaPIGA2 provides high parametrization, manual batch file and command line processing. However, it also offers an extensive graphical user interface for parameter setting, generating and running batch files, following run progress, and manipulating result trees. MetaPIGA2 uses standard formats for data sets and trees, is platform independent, runs in 32 and 64-bits systems, and takes advantage of multiprocessor and/or multicore computers.

Conclusions. The metapopulation Genetic Algorithm resolves the major problem inherent to classical Genetic Algorithms by maintaining high inter-population variation even under strong intra-population selection. Its implementation into a single software with additional stochastic heuristics will allow their rigorous optimization as well as a meaningful comparison of performances among these algorithms. MetaPIGA2 gives access both to high parameterization for the phylogeneticist, as well as to an ergonomic interface and functionalities assisting the non-specialist for sound inference of large phylogenetic trees using nucleotide sequences. MetaPIGA2 is freely available to academics at www.metapiga.org and www.lanevol.org .

CONCLUSIONS

Nous avons présenté MetaPIGA 2.0, spécialement développé pour être capable d'inférer de grandes phylogénies à l'aide de méta-heuristiques, dans un cadre informatique stable, performant, et convivial pour l'utilisateur, et en proposant un vaste choix d'outils de modélisation. Il implémente la plupart des modèles de substitution nucléotidiques (JC, K2P, HKY85, TN93, GTR), ainsi que l'hétérogénéité des taux grâce à une distribution gamma et/ou une proportion d'invariant, et la possibilité de générer une série de réplicats avec un arbre consensus et ses valeurs de support des branches. Il est également l'un des rares logiciels à permettre le partitionnement du jeu données avec des paramètres relatifs à chaque partition (paramètres du modèle, de l'hétérogénéité des taux, proportion d'invariant et longueurs de branches relatives), et la gestion d'un outgroup composé de plusieurs taxa préservé tout au long de l'analyse.

Il offre une excellente alternative aux logiciels basés sur des méthodes Bayesiennes, en proposant 4 méta-heuristiques dont l'implémentation a été adaptée au problème de l'inférence phylogénétique. Nous introduisons une implémentation originale du Simulated Annealing avec un grand choix de courbes de refroidissement et de paramétrages, ainsi qu'une nouvelle version du metaGA en lui adjoignant la possibilité de réaliser de la recombinaison intra- et inter-populations, un paramètre de tolérance permettant d'échapper aux optima locaux pouvant être générés par un consensus, 2 stratégies de gestion des opérateurs de mutation (aveugle ou supervisée), et une condition d'arrêt spécifique basée sur l'erreur moyenne relative entre arbres consensus construits à partir de tous les individus d'une génération. Le metaGA a également été parallélisé, permettant de jouir de bien meilleures performances sur des systèmes multiprocesseurs. Le logiciel permet d'ajouter aisément de nouvelles heuristiques, pouvant utiliser une série d'outils communs tels que les conditions d'arrêt ou les opérateurs de mutations. Un large choix d'opérateurs de mutations est disponible, offrant divers niveau de perturbations, et incluant des classiques (NNI, SPR, TBR) et des inédits (TXS, STS, APRM).

Comme indiqué ci-dessus, nous avons également intégré deux outils d'aide à la décision, l'un permettant de déterminer quelle complexité de modélisation est la plus adaptée au jeu de donnée de l'utilisateur (via les méthodes connues LRT, AIC et BIC mais nécessitant normalement l'utilisation d'un autre logiciel tel que MODELTEST), et l'autre permettant de déterminer quand suffisamment de réplicats ont été produits pour disposer d'un support statistique suffisant sur l'arbre consensus (via une méthode originale utilisant l'erreur moyenne relative entre différents arbres consensus). Nous avons également introduit une méthode aléatoire intermédiaire de génération de l'arbre de départ : le Loose Neighbor Joining.

Au niveau de la convivialité nous proposons un environnement graphique complet qui dispose d'une aide interactive, de l'affichage des arbres et d'un suivi des analyses montrant des courbes d'évolution dynamique de la tâche en cours. Une série d'utilitaires ont été intégrés pour faciliter la vie de l'utilisateur (le TreeViewer, le générateur d'arbres, le constructeur de consensus), ainsi qu'un mode de lancement en « batch » pour les séries d'analyses, avec une interface permettant d'aisément les mettre en place.

Enfin nous tirons parti des dernières technologies en offrant une version 64-bit de MetaPIGA, pouvant travailler avec n'importe quelle quantité de mémoire RAM disponible. Le metaGA n'est pas le seul à profiter d'une machine multiprocesseur ou d'un cluster, la génération de réplicat pouvant être tout aussi aisément parallélisée pour des gains de performances accrus.

Nous pensons donc que MetaPIGA 2.0 offre une ergonomie supérieure à celle de tous les autres logiciels d'inférence phylogénétique que nous connaissons. MetaPIGA 2.0 peut servir de plateforme pour une comparaison rigoureuse (1) des performances de différentes combinaisons de paramètres au sein d'une heuristique, et (2) des différentes heuristiques implémentées (Hill Climbing, Simulated Annealing, GA, et MetaGA). En outre, les performances du metaGA, bien qu'exceptionnelles sous des modèles simples (JC) à intermédiaires (HKY), n'ont pas pu être analysées à ce jour sous des modèles complexes (GTR). L'implémentation d'une hétérogénéité de taux modélisée par une distribution gamma permettra aussi pour la première fois de comparer les performances du metaGA et des autres heuristiques disponibles dans les conditions les plus complexes utilisées sur les jeux de données de grande taille.

CADRE PHYLOGÉNÉTIQUE POUR LA COMPARAISON DE GÉNOMES MULTI-ESPÈCES

INTRODUCTION

Depuis une quinzaine d'années, le nombre de génomes complètement séquencés ne cesse d'augmenter (Liolios, et al., 2006), générant d'immenses quantités de données exploitables et ayant conduit à l'apparition d'une nouvelle discipline en biologie : la génomique comparative (l'étude comparative de la structure et fonction des génomes de différentes espèces). Les buts principaux de cette discipline sont de mieux comprendre comment les différentes espèces ont évolué, quels sont les effets de la sélection sur l'organisation et l'évolution des génomes, ainsi que de déterminer les fonctions des gènes et des régions non-codantes du génome. Les biologistes ont par exemple beaucoup appris sur certaines fonctions de gènes humains en comparant leurs homologues d'autres organismes tels que la souris.

L'imposante masse de données contenue dans les génomes nécessite un haut niveau d'automatisation pour la plupart des méthodes utilisées dans le domaine, et les approches informatiques pour la comparaison de génomes sont un sujet de recherche de plus en plus commun en informatique. De nombreux logiciels et algorithmes ont donc été développés pour aligner des génomes complets et faciliter la comparaison multi-génomes, et des bases de données publiques ont été créées pour l'annotation de ces génomes, pour rassembler les données fonctionnelles et d'expression, ainsi que pour créer des liens entre la taxonomie ou les publications avec les différents projets de séquençage. Beaucoup de ces bases de données traitent un sous-domaine bien précis et sont complémentaires entre elles. Par exemple, la base de données ENSEMBL (Hubbard *et al.*, 2007) fournit une annotation automatisée de séquences métazoaires, identifie des relations d'orthologie et de paralogie via l'estimation d'arbres phylogénétiques de familles de gènes, ainsi que des liens vers des termes d'ontologie de gènes. Une autre base de données, PANTHER, met en relation des fonctions moléculaires et des processus biologiques avec des sous-familles de protéines définies phylogénétiquement (Mi *et al.* 2007). HMDEG quand à elle classifie des millions d'ESTs humains et de souris en catégories de tissus/organes (Pao *et al.* 2006).

Bien qu'on puisse trouver une clé de référence entre ces bases de données, via l'identifiant d'un gène par exemple, il reste fastidieux de rassembler les informations de chacune d'elle pour ne fusse qu'un seul gène. Il serait cependant intéressant de visualiser et comparer ces différentes données pour des familles entières de gènes, voir des groupes plus importants. C'est ce manque de relations et le besoin d'analyses croisées entre bases de données biologiques qui nous a poussés à développer un logiciel qui tirerait parti de ces différentes sources d'informations pour répondre à des questions multicritères sur un ensemble de gènes.

Nous proposons de travailler dans un cadre phylogénétique, en rassemblant en un même caractère les orthologues[24] d'un même gène dans les différentes espèces, et en le plaçant dans la phylogénie sur la branche où il est le plus probable qu'il soit apparu (déterminée en se basant sur les événements de duplication). En observant les espèces ne possédant pas ce caractère et se limitant au sous-arbre descendant de cette branche, nous pourrons également inférer sur quelles branches ce caractère a été perdu. Une fois les gains et pertes de caractères définis, nous pourront les utiliser pour « reconstruire » le contenu du génome des espèces ancestrales. En utilisant les données sur les processus biologiques, les fonctions moléculaires, et les tissus dans lesquels ces gènes sont exprimés, nous pourrons déterminer les fonctions/processus/tissus qui sont statistiquement sur- et sous-représentés au cours de l'évolution (sur chaque branche de la phylogénie). Nous pourrons également observer si des caractères qui ont été gagnés, dupliqués ou perdus sont exprimés spécifiquement dans un seul tissu. Avec ces informations nous pourrons générer des statistiques globales sur l'ensemble des génomes de notre phylogénie au cours de l'évolution, pour les différents domaines que nous venons d'évoquer. Enfin, nous permettrons d'exploiter encore plus cette source d'information en fournissant un système de requêtes complexes, via lequel un utilisateur pourra poser des questions combinant l'identité des gènes, leur présence sur une branche ou leur cartographie (gain/perte), et leurs fonctionnalités (processus biologiques, fonctions moléculaires, données d'expressions).

Notre logiciel va donc permettre d'explorer et d'interroger le contenu de génomes entièrement séquencés et leurs données fonctionnelles associées dans un contexte phylogénétique. Comme notre logiciel va se baser sur les données contenue dans ENSEMBL, nous allons brièvement expliquer comment ils déterminent les relations d'homologie entre gènes, et ce que sont et impliquent les événements de duplication de gènes.

ÉVÉNEMENTS DE DUPLICATION ET DÉTERMINATION DES RELATIONS D'HOMOLOGIE

Les événements de duplication de gènes sont supposés avoir joué un rôle prépondérant dans l'évolution des organismes vertébrés, et il est aujourd'hui évident qu'il s'agit du mécanisme le plus important permettant de générer de nouveaux gènes et de nouveaux processus biochimiques qui ont facilité l'évolution des organismes (Raes & Van de Peer 2003). La duplication des gènes est également importante pour générer plusieurs copies de gènes ayant la même fonction, permettant la production d'une plus grande quantité d'ARN ou de protéines. La duplication des gènes peut avoir différentes conséquences fonctionnelles, comme (1) l'adaptation tissulaire (les 2 copies gardent leurs fonctions mais une des copies acquiert une expression spécifique à un tissu), (2) la perte d'une fonction pour une copie (n'étant plus exprimé et devenant un pseudogène[25]), (3) la spécialisation (par exemple le gène originel avait 2 fonctions et ses 2 copies perdent chacune une fonction différente, l'organisme

[24] **Orthologue** : on parle d'orthologie lorsqu'une relation d'homologie entre une paire de gènes est générée par un événement de spéciation. Voir Figure 4 page 14.

[25] **Pseudogène** : gène dont la séquence est voisine des gènes de structure fonctionnels, mais qui ne s'exprime pas.

est donc toujours capable d'assurer les 2 fonctions mais avec 2 protéines différentes), ou même (4) la création de nouvelles fonctions (une copie conserve la fonction originale et l'autre diverge et acquiert une nouvelle fonction) (Ohno 1970 ; Li 1999). La duplication de gènes est un phénomène fréquent (en moyenne 0.01 fois par gène et par million d'années chez les eukaryotes), mais l'accroissement du nombre de gènes est compensé par une perte de gènes elle aussi assez importante (Lynch & Conery 2000). Nous pouvons rassembler en *familles géniques* des gènes provenant de séquences homologues et ayant évolués grâce à ce processus de duplications.

Dans la base de données ENSEMBL, la prédiction des gènes homologues (orthologues et paralogues) est réalisée par un pipeline où les arbres de gènes phylogénétiques inférés par maximum de vraisemblance jouent un rôle central (Hubbard *et al.*, 2007). Ils tentent de représenter l'histoire évolutive des familles géniques, c'est-à-dire les gènes qui ont divergés d'un ancêtre commun. Lorsque ces arbres de gènes sont réconciliés avec leur arbre d'espèces, leurs nœuds internes sont annotés de manière à distinguer les événements de duplication et de spéciation. Lorsque le nœud ancêtre d'une paire de gènes est un événement de spéciation, ces gènes sont dits orthologues ; et lorsqu'il s'agit d'un événement de duplication, ces gènes sont dits paralogues. Cette approche permet également de déterminer la position dans l'arbre phylogénétique d'espèce sur laquelle les événements de duplication se sont produits (en identifiant l'ancêtre commun le plus proche d'un nœud interne de l'arbre). Nous utiliserons ces arbres de gènes annotés pour rassembler les orthologues en un même caractère et inférer les branches sur lesquelles ces caractères ont été gagnés et perdus.

DÉVELOPPEMENT DE MANTIS

Nous avons baptisé notre logiciel MANTiS, qui en grec signifie « devin, prophète, voyant ». Les grecs, qui avaient fait la connexion entre les pattes levées d'une mante attendant sa proie et les mains d'un prophète en prière, ont utilisé le terme *mantis* pour parler de la « mante religieuse ». MANTiS tiendra donc son rôle de devin en tentant de répondre aux questions de ses utilisateurs, en utilisant une mante religieuse pour symbole !

Pour pouvoir assurer ses fonctions, MANTiS s'appuie sur une base de données mettant en relation plusieurs sources distinctes. Cette base de données reposera principalement sur les génomes séquencés par ENSEMBL et les arbres de gènes construits via leur pipeline. Nous utiliserons les identifiants de gènes d'ENSEMBL, afin d'éviter à l'utilisateur de devoir convertir les gènes qu'il a déjà pu identifier chez ENSEMBL dans un nouveau format. Au début du développement de MANTiS, la version 39 d'ENSEMBL proposait 19 espèces, et n'a depuis cesser de croître, la version actuelle (56) proposant 51 espèces. Nous allons donc mettre en place un pipeline automatisé permettant de rester à jour par rapport à ENSEMBL, afin de pouvoir bénéficier de cet apport régulier de nouvelles données. Pour les processus biologiques et les fonctions moléculaires, nous utiliserons les données de PANTHER[26], qui utilise les

[26] **PANTHER** est une base de données mettant en relation des fonctions moléculaires et des processus biologiques à des sous-familles de protéines définies phylogénétiquement (Mi, et al., 2007)

identifiants *Entrez*[27] pour 4 espèces (l'humain, la souris, le rat et la drosophile). Pour les données d'expression, la base de données d'ENSEMBL garde à jour des lien vers les données de eGenetics (Kelso, et al., 2003) et GNF (Su, et al., 2002) pour l'être humain[28]. Nous utiliserons ces 2 sources de données, ainsi que celles de la base HMDEG[29] (humain également) pour laquelle il faudra convertir les identifiants Unigene[30].

La base de données de MANTiS pourra être consultée via une interface graphique, permettant à la fois d'explorer la phylogénie via une représentation en arbre, et par un système de requêtes simplifiées. La représentation en arbre permettra d'afficher les données sur chaque branche (que ce soit des valeurs comme le nombre de gène gagnés sur une branche, ou un graphique comme l'histogramme des processus biologiques humains surreprésentés sur une branche), avec un système de zoom. Le système de requêtes devra permette de poser un éventail de questions à la base de données qui soit le plus large possible, sans obliger l'utilisateur à en connaître sa structure et sans qu'il ait besoin de notion dans un langage d'interrogation de base de données (comme SQL[31] par exemple). Plusieurs graphes de statistiques générales pourront également être produits, reprenant l'évolution des différents types de données au cours du temps pour l'ensemble de la phylogénie.

[27] **Entrez** est un système global de recherches inter-bases de données permettant d'accéder aux différentes bases données du National Center for Biotechnology Information (NCBI).
[28] **eGenetics** et **GNF** sont des bases de données d'expression de gènes humains annotées selon l'ontologie EVOC. Voir en détails « **Erreur ! Source du renvoi introuvable.** » page 77.
[29] **HMDEG** est une base de données qui classifie des millions d'EST humains et de la souris en catégories de tissus/organes (Pao, et al., 2006)
[30] **Unigene** est une base de données du transcriptome hébergée au NCBI et fournissant pour ses entrées des données d'expression, de similarité avec d'autres protéines, des clones cDNA et la localisation dans le génome.
[31] **SQL** : Structured Query Langage. Le SQL est un langage largement reconnu et répandu. Bien qu'il ne s'agisse pas d'un véritable langage, au sens de C ou de Pascal, SQL peut servir à formuler des questions de façon interactive, mais peut aussi être inséré dans une application sous forme d'instructions de manipulation de données.

MANTIS: A PHYLOGENETIC FRAMEWORK FOR MULTI-SPECIES GENOME COMPARISONS

L'article « *MANTiS: a phylogenetic framework for multi-species genome comparisons* » a été publié en 2007 dans la revue Bioinformatics. Dans cet article, nous présentons le logiciel MANTiS, en passant en revue ses principales fonctionnalités et en décrivant le processus de génération d'un jeu de données. Vous pouvez le consulter sur le site http://www.mantisdb.org.

L'introduction s'attarde tout d'abord sur les différentes sources de données qui seront intégrées dans MANTiS, puis décrit le cadre phylogénétique sur lequel se basera le logiciel, et quelles tâches il sera en mesure d'accomplir. Dans la deuxième partie, « System and methods », nous décrivons le processus de génération des différentes informations contenue dans un jeu de donnée MANTiS. La première section décrit comment les gènes contenus dans les arbres de gènes d'ENSEMBL sont rassemblés en caractères MANTiS selon les événements de duplications, et nous discutons l'intérêt de générer un jeu de données contenant tous ces caractères (intitulé « *with duplications* ») et un autre où les caractères représentent une famille de gène complète (intitulé « *families only* »). Nous y expliquons également comment MANTiS détermine l'ancestralité des caractères, en utilisant la distance moyenne des branches entre les duplications et les gènes ; et comment l'identifiant des caractères MANTiS est déterminé, via la nomination d'un gène principal. La deuxième section décrit comment nous établissons la cartographie des caractères sur la phylogénie (c'est-à-dire en déterminant sur quelle branche un gène est gagné, et sur quelles branches il a éventuellement été perdu), et nous introduisons les 2 types de cartographie disponibles : « *All Changes* » et « *Single Changes* ». La troisième section décrit le processus de reconstruction des génomes ancestraux pour chaque branche de la phylogénie. La quatrième section explique comment les relations avec les processus biologiques et les fonctions moléculaires sont créées avec la base de données PANTHER, et comment MANTiS détermine si la sur- ou la sous-représentation d'une catégorie sur une branche de la phylogénie est statistiquement significative. La cinquième section présente les 3 sources de données que MANTiS utilise pour les données d'expression de gènes, et décrit le filtre appliqué pour obtenir des catégories de gènes exprimés spécifiquement dans un tissu. La sixième partie décrit le système de requêtes élaborées que MANTiS propose pour interroger sa base de données. Nous y décrivons son utilisation et les types de critères disponibles, ainsi que la possibilité de restreindre un jeu de données à un ensemble de gènes obtenu en résultat d'une requête, MANTiS recalculant alors toutes ses statistiques sur base de ce jeu de données réduit. Dans la troisième partie de l'article, « The Mantis Views », nous présentons tour à tour les différentes vues de MANTiS, permettant d'explorer la phylogénie selon un thème choisi (la cartographie des gènes, les processus biologiques, etc.). Dans la quatrième partie, nous utilisons MANTiS pour faire une étude de cas, centré sur 2 exemples concrets de biologie. Nous concluons en récapitulant les innovations apportées par les différentes fonctionnalités de MANTiS, et en proposant de futures extensions avec d'autres sources de données.

TITLE AND ABSTRACT

MANTiS, a phylogenetic framework for multi-species genome comparisons

Raphaël Helaers[1#], Athanasia C. Tzika[1#], Yves Van de Peer[2] & Michel C. Milinkovitch[1*]

Bioinformatics 2008 24(2):151-157

[1] *Laboratory of Evolutionary Genetics, Institute for Molecular Biology & Medicine, Université Libre de Bruxelles,*
12 rue Jeener & Brachet, B6041 Gosselies, Belgium.

[2] *Bioinformatics & Evolutionary Genomics, Department of Plant Systems Biology, Ghent University, VIB, Gent, Belgium*

[#] *The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.*

Received on September 22, 2007; revised and accepted on November 7, 2007

Advance Access publication November 19, 2007

Associate Editor: Martin Bishop

Motivation. Practitioners of comparative genomics face huge analytical challenges as whole genome sequences and functional/expression data accumulate. Furthermore, the field would greatly benefit from a better integration of this wealth of data with evolutionary concepts.

Results. Here, we present MANTiS, a relational database for the analysis of *(i)* gains and losses of genes on specific branches of the metazoan phylogeny, *(ii)* reconstructed genome content of ancestral species, and *(iii)* over- or under-representation of functions/processes and tissue specificity of gained, duplicated, and lost genes. MANTiS estimates the most likely positions of gene losses on the true phylogeny using a maximum likelihood function. A user-friendly interface and an extensive query system allow to investigate questions pertaining to gene identity, phylogenetic mapping, and function /expression parameters.

Availability. MANTiS is freely available at http://www.mantisdb.org and constitutes the missing link between multi-species genome comparisons and functional analyses.

Contact. Michel C. Milinkovitch, mcmilink@unige.ch

MÉTHODOLOGIE ET CONCEPTION DU LOGICIEL MANTIS

Maintenant que nous avons présenté les fonctions principales de MANTiS, nous allons détailler la structure de la base de données, comment fonctionne le système de requêtes, comment les statistiques de représentation sont calculées (pour les processus biologiques, fonctions moléculaires et expression des gènes) et quelles statistiques générales MANTiS peut produire.

Nous avons choisi de développer MANTiS en JAVA, pour les mêmes raisons que celles avancées pour le développement de MetaPIGA 2.0 (voir « Méthodologie et conception du logiciel MetaPIGA 2.0 » page 34). Il utilise une base de données MySQL, un des SGDB libres les plus répandus. La base de données est hébergée au LANE (*Laboratory of Artificial and Natural Evolution*, Université de Genève, Suisse) et de nouveaux jeux de données MANTiS sont créés à chaque nouvelle version d'ENSEMBL, ou lorsque des mises à jours des autres sources de données ont lieu. Un pipeline automatisé permet de créer ces nouveaux jeux de données avec un minimum d'intervention humaine.

BASE DE DONNÉES

Dans la structure de la base de données MANTiS, plusieurs groupes de schémas[32] sont à distinguer :

(1) Schémas externes : les schémas relatifs à une version d'ENSEMBL. Ils sont récupérés chez ENSEMBL, mais seule une partie des tables est utilisée par MANTiS. ENSEMBL structure la base de données de chaque nouvelle version en attribuant un schéma pour chaque espèce (duquel MANTiS utilise la liste des gènes, et les données sur les séquences), un schéma Compara reprenant les données de génomique comparative (duquel MANTiS utilise les arbres de gènes) et un schéma Mart (duquel MANTiS récupère les données d'expression EST et GNF, ainsi que les tables de conversion vers d'autres identifiant, tels que les transcrits ou les identifiants Unigene). Toute la structure des schémas et des tables d'ENSEMBL est conservée pour rester le plus compatible possible, mais les tables inutiles à MANTiS sont vides. Le nom des schémas sont les mêmes que ceux d'ENSEMBL.

(2) Schémas internes : les schémas correspondant aux différents jeux de données accessibles dans MANTiS. Leur nom commence toujours par '*mantis_*' suivit de type (*withDuplications* ou *familiesOnly*), de son identifiant (un chiffre correspondant aux versions d'ENSEMBL) et de sa phylogénie (*A* ou *B*). Par exemple pour le jeu de données correspondant à la version 48 d'ENSEMBL, avec duplications et utilisant la phylogénie A, nous avons le schéma : *mantis_withDuplications_48_A*.

(3) Schéma commun : un schéma commun à tous les jeux de données de MANTiS : *mantis_main*

[32] **Schéma** : dans un système de gestion de base de données (SGBD), description d'une base de données créée au moyen du langage de définition de données proposé par le SGBD (SQL dans notre cas).

Nous utilisons la structure d'ENSEMBL pour stocker les données de séquençage et les arbres de gènes, mais rien ne nous empêche d'intégrer des données d'autres sources. Nous avons développé des scripts permettant de créer une structure de type ENSEMBL et de remplir les tables avec des données provenant de fichiers textes et d'arbres au format Newick. Grâce à cela, nous avons par exemple pu créer un jeu de données MANTiS « Phylome DB » basé sur les données du projet « Human Phylome » (Huerta-Cepas et al. 2007).

Pour une même version d'ENSEMBL (ou de tout autre source de données), plusieurs jeu de données MANTiS sont créés. Comme nous l'avons détaillé dans la section **Erreur ! Source du renvoi introuvable.** (page **Erreur ! Signet non défini.**) nous différencions les jeux de données « *With duplications* » et « *Families only* », mais nous proposons également pour chacun d'eux 2 phylogénies différentes. En effet, dans la « vraie » phylogénie que nous utilisons (voir **Erreur ! Source du renvoi introuvable.** page **Erreur ! Signet non défini.**), la résolution du nœud 35 est sujette à controverse (Blair *et al.* 2002). Nous proposons donc les 2 solutions les plus répandues, que nous avons appelées phylogénie A et B (Figure 18), et produisons un jeu de données MANTiS pour chacune d'elle (la distribution des gains et des pertes étant différente). Nous avons donc pour chaque version d'ENSEMBL un total de 4 jeux de données, 2 avec duplications (selon la phylogénie A ou B) et 2 avec uniquement les familles (également selon la phylogénie A ou B).

Figure 18 – La différence entre les 2 phylogénies proposées par MANTiS. Le placement du nématode (C. Elegans) étant encore incertain (Blair *et al.* 2002), la phylogénie A supporte l'hypothèse d'une branche « Coelomata » (Chordata + Diptera), tandis que la phylogénie B supporte l'hypothèse d'une branche « Ecdysozoa » (Diptera + C. Elegans).

La structure d'un jeu de données MANTiS est détaillée dans le diagramme entité-association ci-dessous (Figure 19). Il reprend le schéma commun à tous les jeux de données (*mantis_main*, les 4 tables qu'il contient sont rassemblées dans le cadre pointillé) et les tables du schéma interne. Pour ne pas surcharger le diagramme, les tables ayant une structure identique ont été regroupées sous un intitulé générique.

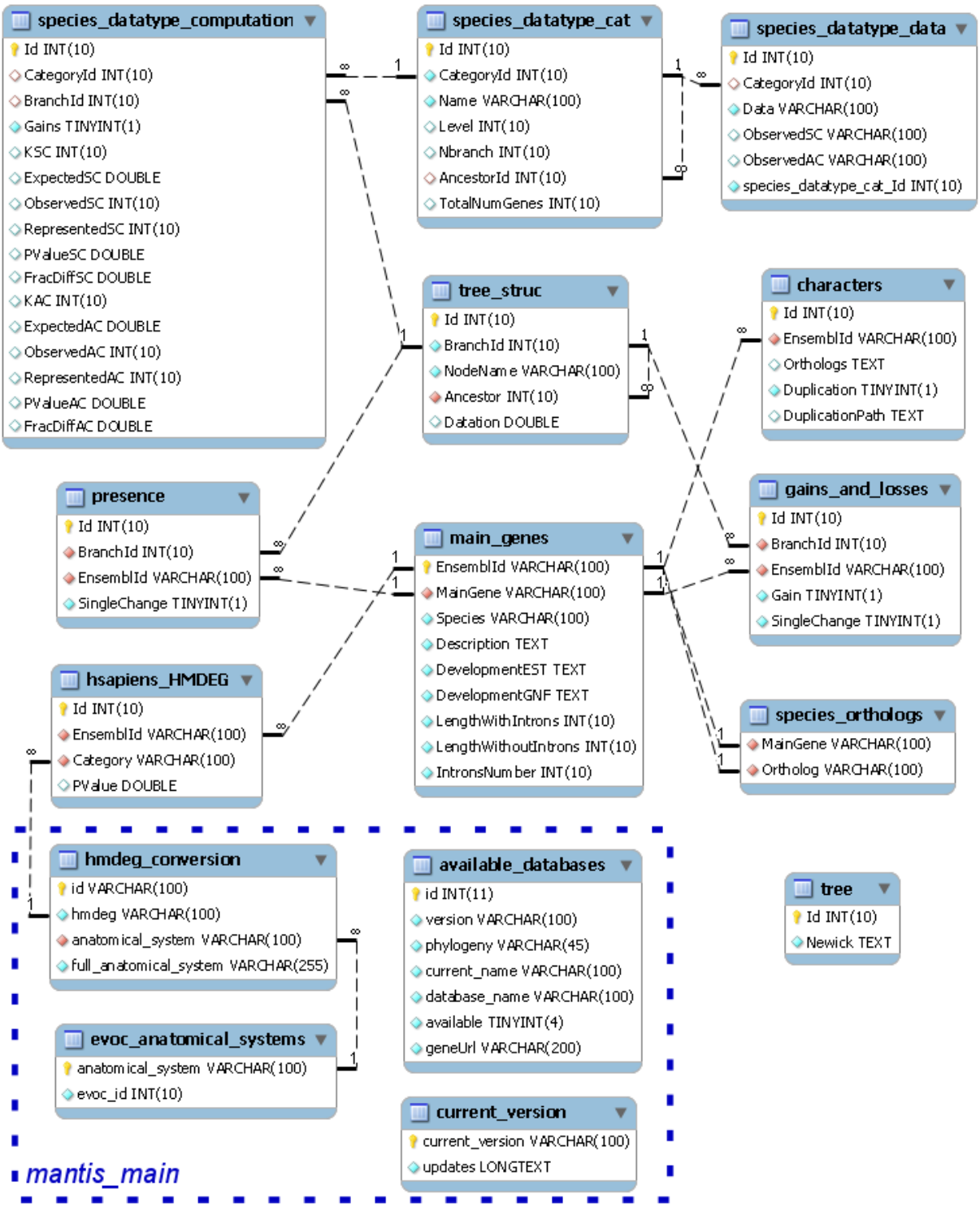

Figure 19 – Diagramme Entité-Association d'une base de données MANTiS.

Le schéma commun, *mantis_main*, contient 4 tables. *available_databases* détermine quels sont les jeux de données accessibles depuis MANTiS, et les schémas (internes et externes) qui lui sont associés. Le champ *geneUrl* permet de stocker l'url d'accès à la page web d'un gène de cette espèce. Il est utilisé dans les fenêtres de type « info-gène » pour créer un lien cliquable vers la base de données d'origine. La table *current_version* contient la version courante de MANTiS et les notes de mise à jour. Elle est accédée au lancement de MANTiS qui comparera sa version à celle indiquée dans cette table. Si elle est différente, MANTiS ne se lancera pas et demandera d'effectuer une mise à jour. MANTiS utilise la table *evoc_anatomical_system* pour uniformiser les données d'expression, les systèmes anatomiques qui y sont repris faisant partie de la norme eVOC[33] (Kruger *et al.* 2007). La table *hmdeg_conversion* fournit l'équivalent des catégories EVOC pour les catégories utilisées par HMDEG.

L'arbre phylogénétique sur lequel MANTiS se base est stocké sous format Newick dans la table *tree*, et détaillé dans la table *tree_struc* où un identifiant unique est attribué à chaque branche (et réutilisé par les autres tables, en suivant les relations 1:N du diagramme). Le champ *Ancestor* définit la branche parente de chaque branche.

La table *main_gene* est centrale, car elle reprend chaque gène du jeu de données (sous son identifiant ENSEMBL) et stocke les différentes données disponibles sur lui (espèce, description, taille, stades de développement). Le plus important étant le champ *MainGene* qui indique quel orthologue de ce gène est considéré comme le gène principal. Nous avons expliqué dans la section **Erreur ! Source du renvoi introuvable.** (page **Erreur ! Signet non défini.**) que nous regroupions tous les gènes orthologues entre eux sous un même caractère. Plutôt que de donner un identifiant spécifique à MANTiS à ce caractère, nous gardons un lien vers la base de données d'origine (donc généralement ENSEMBL) en utilisant l'identifiant du gène appartenant à l'espèce la plus prioritaire (voir **Erreur ! Source du renvoi introuvable.** page **Erreur ! Signet non défini.** pour la liste des priorités). Cette table pourra donc être utilisée pour obtenir le gène principal d'un caractère avec l'identifiant de n'importe quel gène compris dans ce caractère (ou inversement). La table *characters* reprend quand à elle chaque caractère MANTiS, sous l'identifiant de son gène principal, en indiquant si le caractère provient d'une duplication (ou s'il est *de-novo*), si d'autres duplications ont eu lieu avant et la liste complètes des orthologues du gène principale.

Les données de cartographie sont stockées dans les tables *gains_and_losses* et *presence*. Nous y associons l'identifiant du gène principal de chaque caractère avec chaque branche sur laquelle il est présent (*presence)*, gagné (*gains_and_*losses avec *Gain* = 1) ou perdu (*gains_and_*losses avec *Gain* = 0). L'association est faite en « All Changes » et en « Single Changes » (le champ *SingleChange* est mis à 0 ou à 1).

species_datatype_cat, species_datatype_computation, et *species_datatype_data* sont les tables stockant un ensemble de catégories. Il y a un exemplaire de chacune de ces tables pour chaque association *species* + *datatype* disponible. Il y a 4 *datatypes*, comprenant les processus biologiques, les fonctions moléculaires, l'expression de gènes et l'expression spécifique à un

[33] **eVOC** : une ontologie de tissus dans lesquels les gènes humains ou de la souris peuvent être exprimés. Elle est utilisée dans eGenetics et GNF par exemple.

tissu ; les deux premiers étant disponibles chez l'humain, la souris, le rat et la drosophile, et les 2 derniers étant disponibles uniquement pour l'humain. Dans ces trois tables, une catégorie est identifiée par son *CategoryId* unique, générant les relations 1:N représentées sur le diagramme. La table *species_datatype_cat* reprend la structure des catégories (*AncestorId* indiquant la catégorie parente, nous pouvons reconstruire l'arborescence complète dans MANTiS), et le nombre total de gènes dans chacune d'elle. *species_datatype_data* indique quel gène se trouve dans quelle catégorie (champ *data*) et sur quelles branches le gène peut être observé (les champs *ObservedAC* et *ObservedSC* contiennent une chaîne de 0 et de 1, représentant chaque branche de l'arbre avec un 1 si le gène y est présent et 0 sinon, respectivement en « All Changes » et en « Single Changes »). Les statistiques de représentation de chaque catégorie sont calculées pour chaque branche, et stockées dans *species_datatype_computation*. Comme les catégories sont spécifiques à une espèce, elles utilisent toujours les identifiants des gènes de cette espèce. Comme il est possible qu'un gène de cette espèce fasse partie d'un caractère pour lequel il n'est pas le gène principal, une table de conversion permet de passer aisément de l'un à l'autre. Nous avons donc une table *species_orthologs* pour chaque espèce associée à un ensemble de catégories (donc *hsapiens, mmusculus, rnorvegicus* et *dmelanogaster*).

REQUÊTES

Nous avons introduit le système de requêtes proposé par MANTiS dans l'article qui lui est dédié. Dans cette section, nous allons entrer plus en profondeur dans les critères qui peuvent être associées à ces requêtes.

Si une requête inclus plusieurs « statements », chacun d'eux est exécuté séparément, et les résultats sont groupés selon les opérateurs logiques sélectionnés : l'intersection pour l'opérateur « *and* », la différence asymétrique pour l'opérateur « *and not* », l'union pour l'opérateur « *or* », et la différence symétrique pour l'opérateur « *xor* » (voir Figure 20). Les priorités gérant les opérateurs sont également définies par l'utilisateur (par exemple, si nous avons 3 statements A, B et C, la requête « (A *and* B) *or* C » n'est pas équivalente à la requête « A *and* (B *or* C) »).

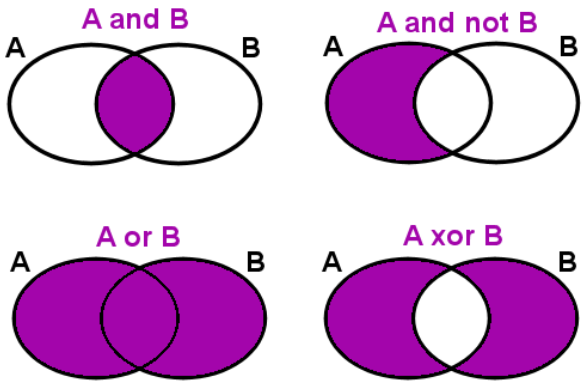

Figure 20 – Opérateurs logiques permettant de lié les « statements » d'une requête.

De plus, les actions « *Count mapping* » / « *Count functions* » ajoutent une condition de comptage (c'est-à-dire un opérateur de comparaison et un seuil ; par exemple « > 2 ») au critère de cartographie/fonction : les occurrences ayant le même type de cartographie/fonction sont groupés et comptés, et seuls les gènes correspondants à ce critère sont affichés dans la fenêtre de résultat. Cette fonctionnalité permet d'identifier aisément, par exemple, les gènes qui ont été gagnés sur la branche *X* (disons, la racine de la lignée des mammifères) et perdu moins de 2x au sein des mammifères. La requête peut être affinée à l'aide de la commande « *display* » (affectant les champs « *Gene* » et/ou « *Mapping* » et/ou « *Branch* » et/ou « *Function* ») : par exemple, si un gène est perdu sur plus d'une branche, choisir d'afficher le champ branche retournera un résultat pour chaque branche sur laquelle le gène a été perdu au lieu d'un seul résultat avec uniquement l'information de perte du gène. Inversement, la sélection d'un champ « *Group & Count* » fusionnera les résultats ayant la même information dans les champs restants (et affichera leur nombre à la place) : par exemple, si l'utilisateur décide d'afficher les gènes, la cartographie, les branches et les fonctions, et que les gènes doivent être « *Group & Count* », la fenêtre de résultats inclura une liste des combinaisons uniques de branches, gains ou pertes, et fonctions avec le nombre des gènes associés à chaque combinaison (par exemple, 83 gènes perdus avec la fonction *x* sur la branche *y*).

La Figure 21 montre un exemple d'une requête complexe réalisée dans MANTiS : « Liste les gènes, parmi les 10 000 identifiants *Entrez* que je fournis, qui sont spécifiquement exprimés dans le système nerveux (données humaines EST), sont associées à un processus de type développemental, sont gagnés entre l'origine des vertébrés et l'origine des euthériens, et sont présent chez l'humain ainsi que chez mes 2 espèces modèle de laboratoire (par exemple la souris et le chien). ».

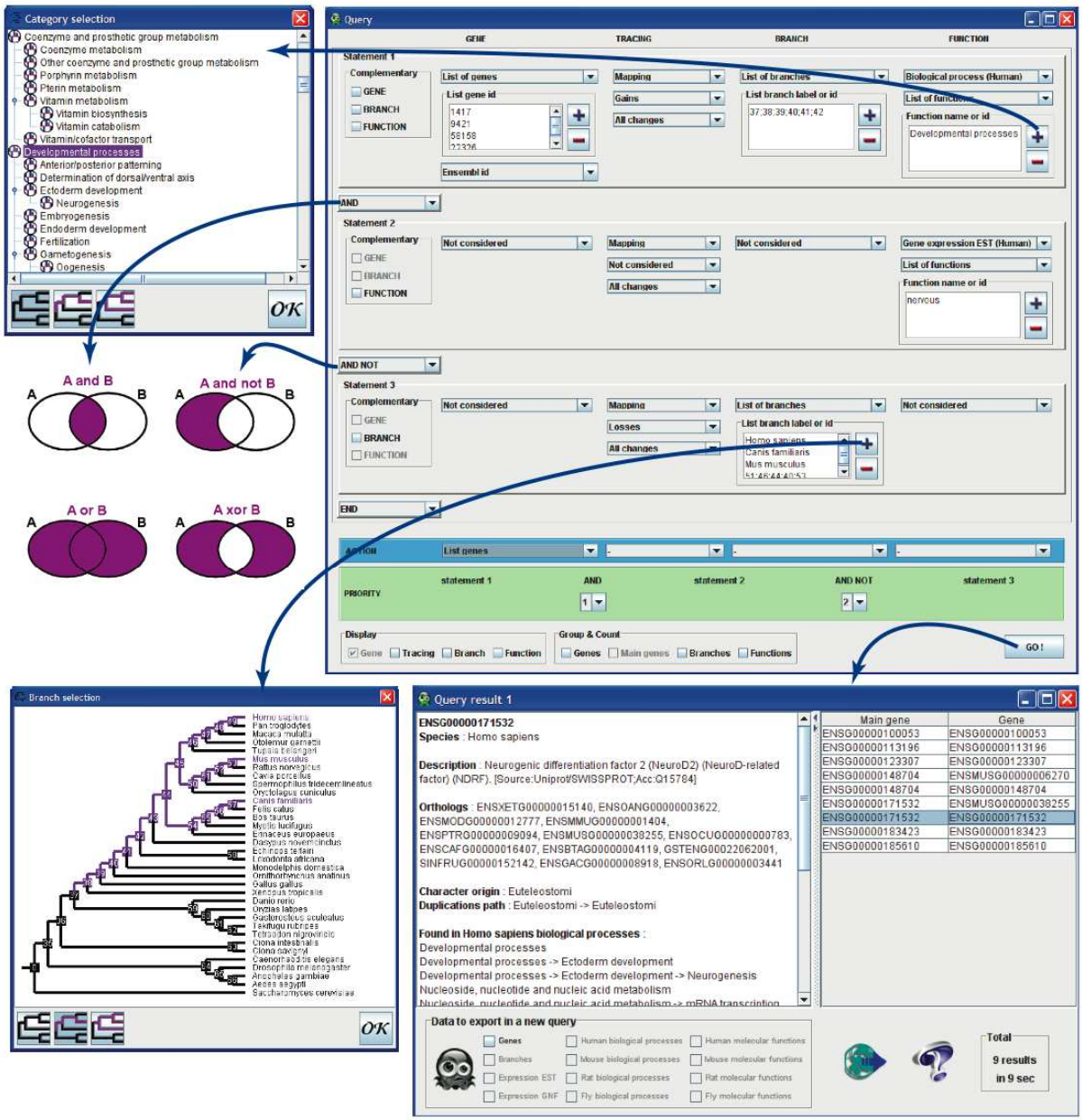

Figure 21 – Exemple d'une requête complexe dans MANTiS.

La requête est construite de la manière suivante : Liste les gènes, parmi les 10 000 identifiants Entrez que je fournis (partie gauche du *statement* 1) , qui sont assignés à la catégorie « *Developmental* » dans la liste des « *Biological Processes* » humains (partie droite du *statement* 1) et qui ont été gagnés chez l'ancêtre commun le plus proche (MRCA – Most Recent Common Ancestor) des vertébrés (branche 37) ou des tétrapodes (branche 38) ou des amniotes (branche 39) ou des mammifères (branche 40) ou des thériens (branche 41) ou des euthériens (branche 42) (partie centrale du *statement* 1). Parmi les gènes retournés en résultat, je veux voir seulement ceux qui sont également assignés (selon les données EST humaines) à la catégorie « *Nervous* » dans la liste des « *Gene expression* » (*statement* 2, lié avec un « AND » au *statement* 1). Parmi les gènes retournés en résultat, je veux seulement voir ceux qui sont présents chez l'humain, la souris et le chien (*statement* 3). Le *statement* 3 nécessite d'identifier les branches sur lesquelles le gène n'a pas été perdus, par conséquent nous utilisons un opérateur « AND NOT » pour lié les résultats des *statements* 1 et 2 avec le *statement* 3. Un *statement* 3 alternatif aurait pu utiliser le champ « *Presence* » (au lieu du champ « *Mapping -> Loses* ») et de lister « *Homo Sapiens* », « *Mus Musculus* » et « *Canis Familiaris* » dans le champ « *Branch* » ; et un opérateur « *AND* » entre le *statement* 2 et 3 aurait du être utilisé. La sélection des fonctions et des branches est facilitée par les outils graphiques « *Category selection* » (en haut à gauche) et « *Branch selection* » (en bas à gauche). Les 9 gènes trouvés par la requête rencontrent tous les critères et peuvent être exportés ou utilisés dans une nouvelle requête.

STATISTIQUES DE REPRÉSENTATION D'UNE CATÉGORIE

La significativité statistique de la sur- ou sous-représentation d'une catégorie est calculée sur base de la distribution des gènes de l'espèce référence en catégories. Par exemple, une catégorie C est sur-représentée en gains (ou pertes) lorsque $k(C)$, le nombre observé de gènes gagnés (ou perdus) dans la catégorie C, est plus grand que $p(C)K$, le nombre attendu d'événements correspondants (où $p(C)$ est la proportion de gènes dans la catégorie C pour l'espèce référence, et K est le nombre total de gains (ou pertes) sur la branche considérée). La significativité statistique est déterminée par le calcul d'une *p-value* qui suit une statistique binomiale (sous hypothèse *null*, le nombre de gènes associés à C est distribué binomialement avec un paramètre de probabilité $p(C)$) :

$$p - value = \sum \binom{K}{k} p(C)^k (1 - p(C))^{K-k} \qquad \text{Eq.32}$$

Où la somme va de $k(C)$ à K dans le cas d'une sur-représentation (c'est-à-dire lorsque le nombre de gains/pertes observé est plus grand que celui attendu sous hypothèse *null*), et de 0 à $k(C)$ dans le cas d'une sous-représentation. Toutes les catégories ayant une *p-value* < 0,05 sont considérées « *significativement* » sur- ou sous-représentées (Mi *et al.* 2007).

Pour calculer les *p-value* en utilisant une distribution binomiale, MANTiS fait appel à la librairie Colt du CERN, qui utilise l'intégrale de la fonction Beta incomplète pour l'approximer.

STATISTIQUES GÉNÉRALES

MANTiS peut générer toute une série de statistiques pour tous les gènes sur l'ensemble de la phylogénie. Ces statistiques peuvent être relatives au contenu du génome, aux processus biologiques, aux fonctions moléculaires, à l'expression des gènes, et à l'expression dans un tissu spécifique. Tous les graphiques ont une ligne du temps (exprimée en « million years ago ») comme abscisse, les points à l'extrême droite étant associés aux branches terminales de l'arbre (les espèces actuelles donc). Chaque graphique peut également présenter les données en fréquences cumulées ou non.

Les graphiques statistiques reprenant les données de contenu de génomes peuvent afficher en ordonnée le nombre de gènes, la taille totale des génomes (avec ou sans introns), la longueur moyenne des gènes (avec ou sans introns), le nombre total ou moyen d'introns, la taille totale ou moyenne des introns, ou encore le taux cartographie/longueur de branche. Ces données peuvent être affichées pour les gènes présents, gagnés ou perdus sur chaque branche, en « All Change » ou en « Single Change ». Chaque point du graphique représente la donnée choisie sur l'une des branches de la phylogénie, avec donc son âge en abscisse et la valeur de la donnée en ordonnée. Les points sont reliés par des lignes vertes représentant les relations dans la phylogénie. Il est possible de mettre en évidence une partie de la phylogénie dans le graphique, par exemple toutes les branches qui partent de l'humain et mènent à la racine de l'arbre. Il est possible d'afficher la liste des gènes associés à un point, et d'exporter le graphique sous forme d'image associé à un fichier reprenant les données permettant de le redessiner dans un autre logiciel (Excel par exemple). Par exemple, la Figure 22 ci-dessous représente l'évolution du nombre de gènes présents sur chaque branche de la phylogénie au cours du temps. La ligne rouge passe par chaque branche allant de la racine de l'arbre à la branche Homo Sapiens.

Les graphiques sur les processus biologiques, les fonctions moléculaires et l'expression des gènes contiennent tous les mêmes types d'information. L'utilisateur commence par fixer les critères « *All Changes* » ou « *Single Changes* », gains ou pertes, la base de données source (PANTHER uniquement pour les processus biologiques et les fonctions moléculaires ; EST, GNF ou HMDEG pour les données d'expression), et l'espèce de référence (humain, souris, rat ou drosophile ; uniquement humain pour les données d'expression). Il choisit ensuite la catégorie qui l'intéresse et peut afficher en ordonnée, pour chaque branche, le nombre de gènes contenus dans cette catégorie (selon les critères choisis), le nombre de gènes normalisé, ou la *p-value*. Il peut également afficher le nombre moyen de catégories de 1er ou de dernier niveau représentées sur chaque branche. Dans ce cas, chaque point est la moyenne d'un histogramme dont la hauteur de chaque colonne représente le nombre de gène ayant un nombre donné de catégorie. Cet histogramme peut être affiché (et exporté) en cliquant sur un point du graphique. Par exemple, la Figure 23 ci-dessous montre l'évolution de la p-value de la catégorie « Alimentary » en expression de gènes.

Les graphiques de spécificité de tissu sont du même type que ceux d'expression des gènes, mais seules les catégories de premier niveau sont disponibles et il n'y a que 2 types de données affichables en ordonnées : le nombre de gènes par catégorie, et le nombre de gènes normalisé.

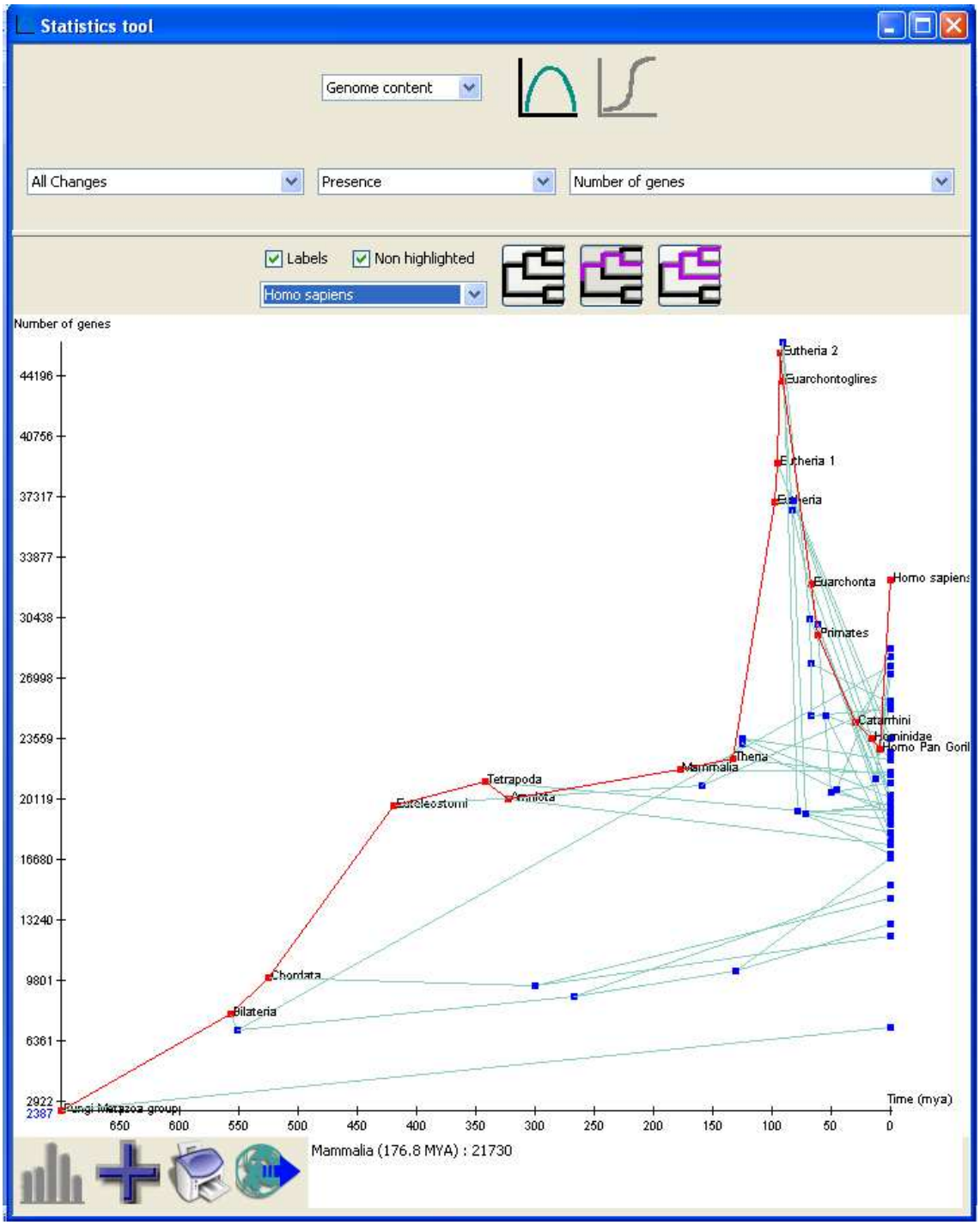

Figure 22 – Nombre de gènes présents sur chaque branche de la phylogénie au cours du temps.

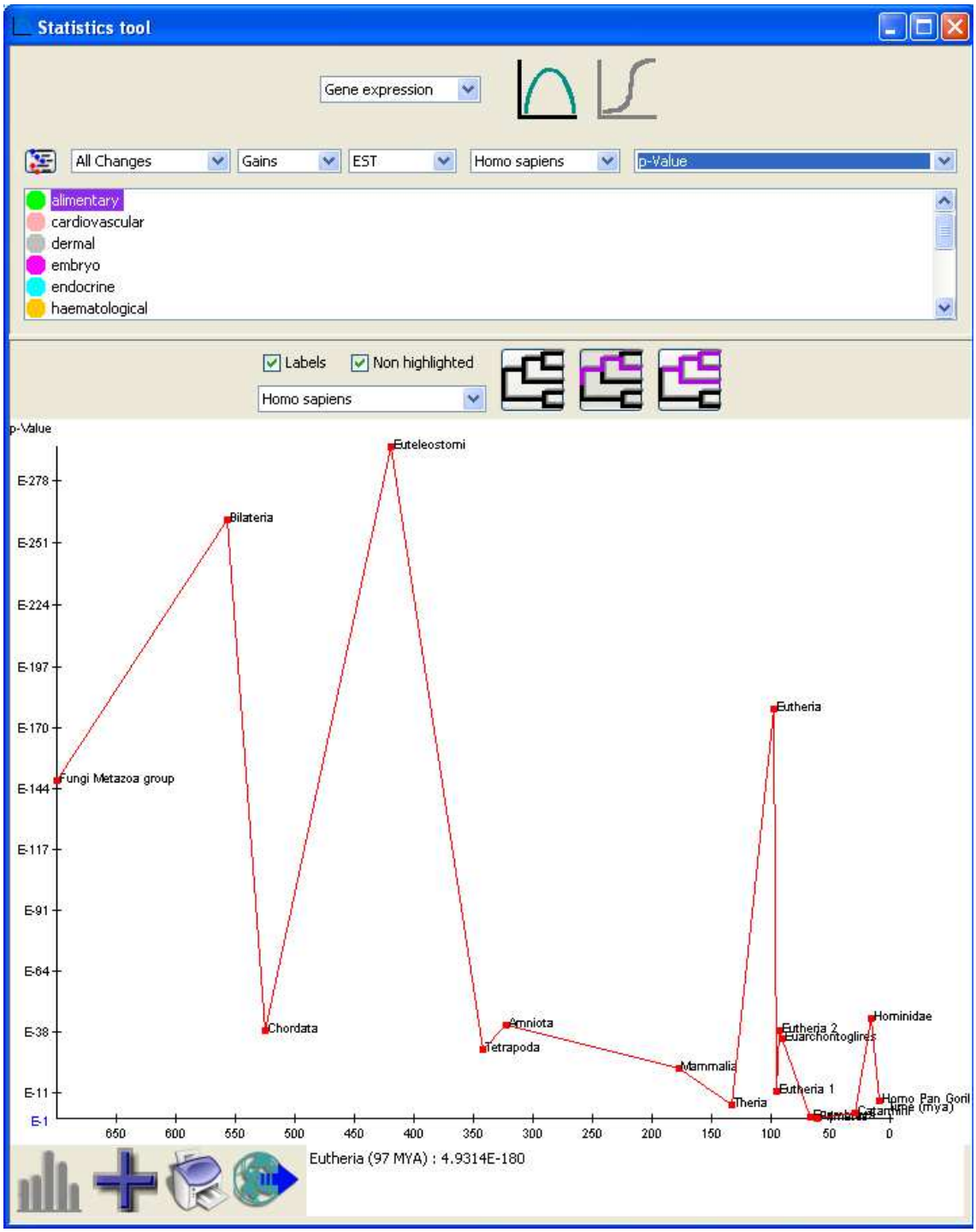

Figure 23 – Distribution des *p-values* sur chaque branche pour la catégorie "Alimentary" en expression de genes. Plus la *p-value* est petite (c'est-à-dire significative), plus le pic est important.

Un dernier type de graphique en 3 dimensions permet d'avoir une visualisation de la distribution de la fréquence des pertes (Figure 24). Les branches où les gènes ont été gagnés sont reprises en abscisse (X), les branches où les gènes ont été perdus sont reprises en ordonnée (Y), et leur nombre est en cote (Z). La hauteur des colonnes indique donc le nombre de gènes qui ont été gagnés sur la branche en X et qui ont ensuite été perdus sur la branche en Y. Une colonne peut être sélectionnée, la colorant en mauve ainsi que les 2 branches associées, et la liste des gènes qu'elle englobe peut être affichée (et exportée). Le graphique complet peut également être exporté (sous forme d'image avec un fichier Excel reprenant l'ensemble de ses données). Un bouton permet d'afficher un histogramme reprenant uniquement la distribution des pertes pour les gènes gagnés sur la branche sélectionnée. Il est également possible de sélectionner une branche en X (Y), et de colorer en rouge (bleu) la distribution des pertes (gains) pour les gènes gagnés (perdus) sur cette branche.

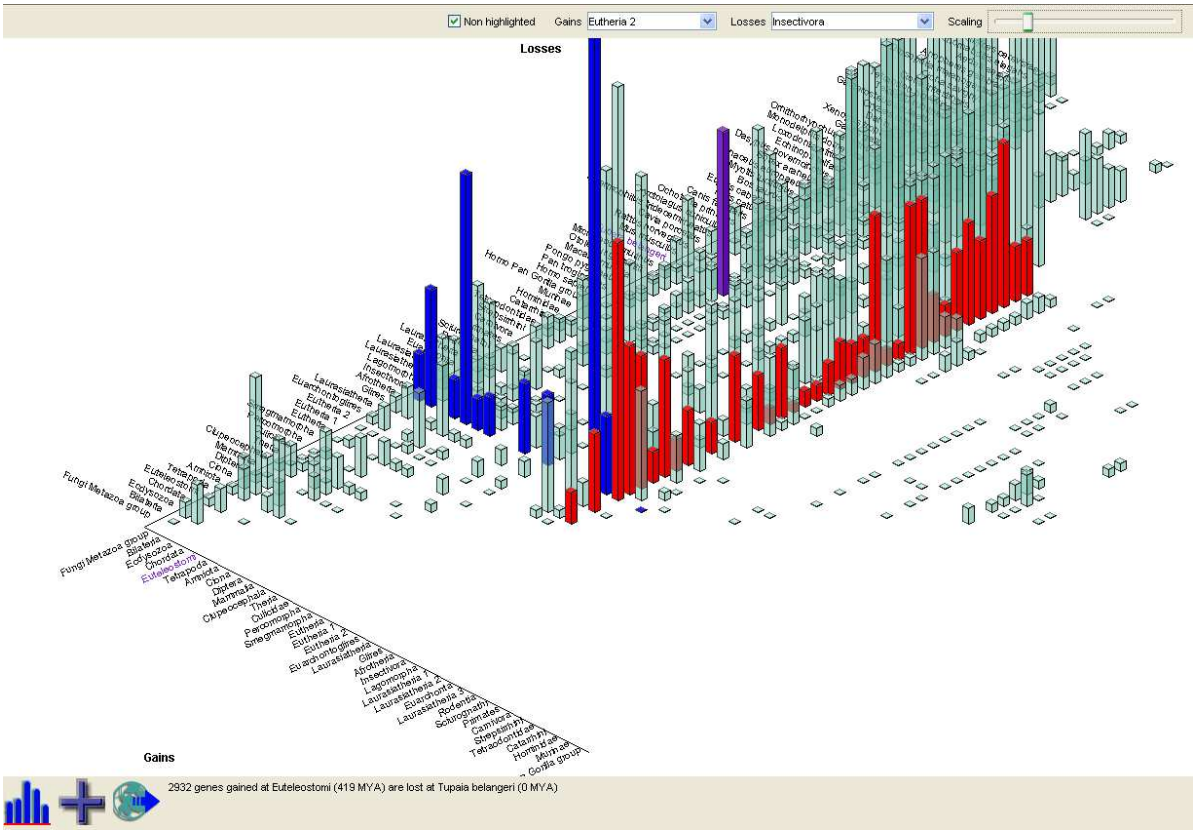

Figure 24 – Graphe de distribution de la fréquence des pertes. Le slider « scaling » permet de changer l'échelle des colonnes, afin de mieux visualiser les plus hautes ou de mieux différencier les plus petites. Dans cet exemple, la colonne mauve a été sélectionnée, indiquant que 2932 gènes ont été gagnés sur la branche Euteleostomi (419 MYA) puis perdus sur la branche Tupaia belangeri (0 MYA). Appuyer sur le bouton « + » ouvrira une fenêtre avec la liste de ces gènes. Appuyer sur le bouton avec les colonnes bleues ouvrira un histogramme avec la distribution des pertes des gènes gagnés sur la branche Euteleostomi. La branche Eutheria 2 a été sélectionnée dans la liste « Gains », colorant en rouge la distribution des pertes des gènes gagnés sur Eutheria 2. La branche Insectivora a été sélectionnée dans la liste « Losses », colorant en bleu la distribution des gains des gènes perdus sur Insectivora. Il est possible de masquer toutes les colonnes restées en vert en décochant la case « Non highlighted ».

UTILISATION DE MANTIS POUR ANALYSER L'ÉVOLUTION DES GÉNOMES

Nous avons introduit MANTiS, et la vaste gamme d'analyses de génomique comparative qu'il permet d'effectuer, en associant cartographie des gains et des pertes, contenu de génome, fonctions moléculaires, processus biologiques et expression des gènes. Nous allons maintenant présenter trois publications, qui grâce une utilisation ciblée de MANTiS ont permis de mettre en exergue des résultats intéressants sur l'évolution des génomes.

La première publication, « *Mapping gene gains and losses among metazoan full genomes using an integrated phylogenetic framework* » (Athanasia C. Tzika, Raphaël Helaers & Michel C. Milinkovitch 2009), compare MANTiS à ses « concurrents » les plus populaires, les bases de données d'orthologues que sont INPARANOID (O'Brien, Remm, and Sonnhammer 2005), MULTIPARANOID (Alexeyenko et al. 2006), ORTHOMCL (Li, Stoeckert, and Roos 2003), et ROUNDUP (Deluca et al. 2006). Nous y verrons que la cartographie des caractères réalisée par MANTiS est bien plus robuste, et génère moins de faux positifs et faux négatifs. Nous y validons ensuite MANTiS, en ré-analysant des familles de gènes connues via des publications. Nous montrerons que les résultats obtenus par MANTiS sont fiables, tout en étant obtenu avec bien plus de facilité, et qu'il nous permet même d'affiné les résultats précédemment publiés.

Avec la seconde publication, « *2x genomes—depth does matter* » (Michel C. Milinkovitch, Raphaël Helaers, Éric Depiereux, Athanasia C. Tzika & Toni Gabaldon 2009), nous tenterons d'illustrer l'impact de l'utilisation de nombreuses espèces séquencées avec trop peu de précision (c'est-à-dire les génomes n'ayant qu'une couverture de 2x), via une analyse de l'évolution des gains et des pertes dans la phylogénie réalisée avec MANTiS. Nous montrerons que l'introduction dans la phylogénie des données de 14 euthériens séquencés avec une faible couverture génère un pic artéfactuel dans le nombre de gains sur la branche des euthériens, suivit d'un nombre important de pertes. Il devient donc très difficile de différencier les artefacts des vrais changements dans les génomes euthériens.

Dans la troisième publication, « *Historical constraints on vertebrate genome evolution* » (Michel C. Milinkovitch, Raphaël Helaers & Athanasia C. Tzika), nous utiliserons MANTiS pour générer plusieurs graphiques associant le nombre moyen de tissus dans lesquels les gènes sont exprimés au cours de l'évolution. Grâce à eux, nous montrerons que la spécificité des tissus dans lesquels un gène est exprimé est contrainte par l'âge de l'origine de ce gène.

MAPPING GENE GAINS AND LOSSES AMONG METAZOAN FULL GENOMES USING AN INTEGRATED PHYLOGENETIC FRAMEWORK

Nous avons publié en 2009 un chapitre du livre « Gene and Genome Duplication » (Wiley-Blackwell 2009, David Liberles eds.) : « *Mapping gene gains and losses among metazoan full genomes using an integrated phylogenetic framework* ». Nous y comparons MANTiS à d'autres bases de données d'orthologues, qui se basent uniquement sur des « Best Reciprocal Hits » Blast pour inférer les gènes orthologues. Nous montrons que grâce à l'utilisation des phylogénies des familles géniques, MANTiS génère une cartographie des caractères bien meilleure. Ensuite, nous utilisons MANTiS pour reproduire 2 exemples de génomique comparative publiés, afin de montrer la facilité et la précision avec laquelle MANTiS permet de retrouver les mêmes résultats, allant même jusqu'à les améliorer.

Ma contribution a été d'extraire et de formater les données des bases de données d'orthologues pour qu'elles puissent être comparées à celles de MANTiS. J'ai également contribué à l'élaboration des requêtes permettant de simuler les exemples publiés.

Le texte commence par introduire les corrélations entre le génome et l'évolution du phénotype, ainsi que l'implication des nombreux événements de duplications chez les eukaryotes, et l'importance d'étudier ces phénomènes dans un cadre phylogénétique. Il présente ensuite MANTiS (la manière dont il organise ses données et la procédure de cartographie des caractères) et le compare à des bases de données d'orthologues populaires (InParanoid, MultiParanoid, OrthoMCL, et RoundUp). La cartographie des caractères générés par MANTiS est comparée aux informations similaires extraites des 4 autres bases de données citées, et nous montrons que ces dernières produisent énormément de faux positifs et de faux négatifs par rapport à MANTiS, au niveau des gènes gagnés entre l'ancêtre commun le plus proche des mammifères et l'ancêtre commun le plus proche de la souris du rat et de l'humain (les 3 espèces couvertes par InParanoid, MultiParanoid, OrthoMCL et RoundUp). La section suivante reproduit ensuite 2 exemples publiés (relatifs (1) aux gains de gènes spécifiques chez la poule et (2) aux gains de gènes impliqués dans le développement de la crête), mais en utilisant MANTiS, montrant qu'il peut réaliser ces analyses de génomique comparative facilement et avec précision, alors qu'elles étaient fastidieuses dans le passé. MANTiS confirme avec aisance les résultats publiés, et en affine même certains grâce à son assignement d'orthologues/paralogues plus élaboré (basé sur des phylogénies) que celui qui avait été utilisé (utilisation de recherches BLAST). La conclusion discute de l'importance d'identifier et de mapper les changements dans le contenu des génomes ayant des implications fonctionnelles en se basant sur une phylogénie robuste, comme le fait MANTiS, ainsi que de la nécessité de disposer de séquences précises et bien annotées pour les génomes séquencés.

TITLE AND ABSTRACT

Mapping gene gains and losses among metazoan full genomes using an integrated phylogenetic framework

Athanasia C. Tzika[1,2], Raphaël Helaers[3] & Michel C. Milinkovitch[1]

[1] *Laboratory of Natural & Artificial Evolution (LANE), Department of Genetics & Evolution, Sciences III, Quai Ernest Ansermet 30, 1211 Geneva-4, Switzerland*

[2] *Evolutionary Biology & Ecology, Université Libre de Bruxelles, Av. F.D. Roosevelt 50, B-1050 Brussels, Belgium*

[3] *Department of Biology & Department of Mathematics, FUNDP, rue de Bruxelles 61, 5000 Namur, Belgium*

Comparative genomics suffers from the lack of integrative tools embedded into a phylogenetic framework. The recently-developed MANTiS relational database (www.mantisdb.org) integrates phylogeny-based orthology/paralogy assignments with functional and expression data into an explicit phylogenetic framework, allowing users to explore phylogeny-driven (focusing on any set of branches), gene-driven (focusing on any set of genes), function/process-driven, and expression-driven questions. Here, we show that such integrative methods provide much improved mapping of gene gains than popular databases of orthologs (InParanoid, OrthoMCL, RoundUp). Furthermore, using published examples pertaining to *(i)* gains of chicken specific genes, and *(ii)* gains of genes involved in neural crest development, we demonstrate that the MANTiS relational database allows to easily and accurately perform comparative genomic analyses that were very tedious in the past.

2X GENOMES—DEPTH DOES MATTER

L'article « *2x genomes—depth does matter* » a été soumis en 2009 et accepté en février 2010 dans la revue Genome Biology. Dans cet article, nous utilisons MANTiS pour analyser l'évolution des gains et des pertes dans la phylogénie, montrant le danger d'utiliser de nombreuses espèces séquencées avec trop peu de précision. Ma contribution porte sur la cartographie du contenu du génome et les analyses des duplications. Lorsqu'il sera publié, l'article sera accessible sur http://www.mantisdb.org

L'article commence par introduire le contexte dans lequel le séquençage de faible précision (2x) peut affecter l'inférence de la cartographie des gains et des pertes. D'un côté, le choix des espèces séquencées suit une logique biomédicale (choix d'espèces proches de l'homme pour aider à comprendre le fonctionnement de la biologie humaine) plutôt que phylogénétique, résultant au séquençage de nombreux mammifères, contre extrêmement peu d'oiseaux, d'amphibiens ou de reptiles par exemple, créant un biais lorsqu'il faut inférer une phylogénie. D'autre part, il a été décidé de séquencer 24 mammifères avec une faible couverture (2x) pour augmenter la puissance statistique de détection des régions conservées pouvant avoir un intérêt biomédical. L'impact de ces génomes à faible couverture dans l'inférence des gains et des pertes va être déterminé en analysant la complexification des génomes eucaryotes à travers les duplications de gènes. Nous détaillons notre méthode d'analyse dans les résultats : nous avons généré des jeux de données MANTiS pour les versions 39 à 48 d'ENSEMBL, ainsi que pour le projet PhylomeDB, puis générer des graphiques montrant l'évolution du nombre de gains (et de pertes) de caractères au cours du temps. La version 39 d'ENSEMBL ne contient pas de génomes avec une faible couverture, mais des espèces séquencées à faible couverture ont été petit à petit ajoutées jusqu'à la version 48, et PhylomeDB ne contient que des espèces séquencées avec une bonne couverture, nous permettant d'observer l'évolution des différents graphiques générés. Dans les dernières versions d'ENSEMBL, il en ressort une impressionnante explosion du nombre de gains lors l'apparition des euthériens, l'ancêtre commun le plus proche des espèces à faible couverture. De plus, la plupart de ces caractères gagnés sont massivement reperdus un peu plus tard chez ces mêmes espèces. Nous discutons ensuite de différents scénarios pouvant expliquer ce pic de gains (suivi de ces pertes massives), qui est très probablement un artefact généré par l'intégration des génomes à faible couverture dans la phylogénie. Nous testons d'abord si ce pic pouvait être causé par une mauvaise phylogénie après l'intégration des afrothériens et des xénarthres. Ensuite, suivant une autre approche nous comparons le taux d'ambiguïté dans les séquences des différentes espèces séquencées. Nous avons également vérifié que les génomes à faible couverture contribuaient significativement plus à la génération de topologies douteuses pour les arbres de gènes d'ENSEMBL. Enfin, nous avons simulé le même taux d'ambiguïtés pour les espèces concernées dans les données de PhylomeDB (qui n'en contenaient pas, et ne présentait pas ce pic), et nous avons pu observer qu'un pic semblable apparaissait alors sur les mêmes branches. Nous faisons suivre ces résultats par une discussion sur l'intérêt d'ajouter des espèces hors des mammifères pour améliorer la fiabilité des analyses, tout en insistant sur le fait que les artefacts dans les gains et pertes de gènes seront toujours présents tant que les génomes à faible couvertures ne seront pas reséquencés avec une meilleure précision. Nous terminons l'article par une section « matériels et méthodes » où nous décrivons la manière dont nous

avons intégré les données de PhylomeDB à MANTiS et détaillons l'algorithme développé pour réconcilier les arbres de gènes avec l'arbre d'espèces.

TITLE AND ABSTRACT

2X genomes — Depth does matter

Michel C. Milinkovitch[1], Raphaël Helaers[2], Éric Depiereux[2], Athanasia C. Tzika[1,3] & Toni Gabaldon[3]

[1] *Laboratory of Artificial & Natural Evolution (LANE), Dept. of Genetics & Evolution Sciences III, 30, Quai Ernest-Ansermet, 1211 Genève 4, Switzerland;*

[2] *Department of Biology, FUNDP, 5000 Namur, Belgium;*

[3] *Dpt of Evolutionary Eco-Ethology, ULB, 1050 Brussels, Belgium;*

[4] *Centre de Regulació Genòmica (CRG), Dr. Aiguader, 88. 08003, Barcelona, Spain;*

Background: Given the availability of full genome sequences, mapping gene gains, duplications, and losses during evolution should theoretically be straightforward. However, this endeavor suffers from overemphasis on detecting conserved genome features, which in turn has lead to sequencing multiple eutherian genomes with low coverage rather than fewer genomes with high-coverage and evener distribution in the phylogeny. Although limitations associated with analysis of low coverage genomes are recognized, they have not been quantified.

Results: Here, using recently-developed comparative genomic application systems, we evaluate the impact of low-coverage genomes on inferences pertaining to gene gains and losses when analyzing eukaryote genome complexification through gene duplication. We demonstrate that, when performing inference of genome content evolution, low-coverage genomes generate not only a massive number of false gene losses, but also striking artifacts in gene duplication inference, especially at the most recent common ancestor of low-coverage genomes. We show that the artifactual gains are caused by genome sequence low coverage *per se* rather than by the increased taxon sampling in a biased portion of the species tree.

Conclusions: We argue that it will remain difficult to differentiate artifacts from true changes in modes and tempo of genome evolution until better homogeneities in both taxon sampling and high-coverage sequencing are met. This is important for broadening the utility of full genome data to the community of evolutionary biologists, whose interests go well beyond widely-conserved physiologies and developmental processes/patterns as they seek to understand the generative mechanisms underlying biological diversity.

HISTORICAL CONSTRAINTS ON VERTEBRATE GENOME EVOLUTION

L'article « *Historical constraints on vertebrate genome evolution* » a été publié en 2009 dans la revue Genome Biology & Evolution. Dans cet article, nous utilisons MANTiS pour montrer que l'âge de l'origine des gènes contraint la spécificité des tissus où ces gènes sont exprimés. Ma contribution a consisté à générer les statistiques d'expression sur l'ensemble des génomes, et d'implémenter la possibilité de restreindre l'arbre et les jeux de données aux seuls caractères provenant d'une duplication (et non *de novo*). L'article est accessible sur le site http://www.mantisdb.org.

L'article commence par introduire le fait que les contraintes liées à l'expression des gènes dupliqués peuvent être mis directement en relation avec l'âge de leur origine, et qu'une telle analyse nécessite d'intégrer les événements de duplication et les données d'expression dans un même cadre phylogénétique, ce que fait MANTiS. Nous présentons ensuite les résultats, en commençant par détailler les données utilisées dans MANTiS et sa méthode de cartographie des caractères, puis les 3 sources de données d'expressions sur lesquelles MANTiS se base. Nous expliquons comment nous avons généré plusieurs graphiques montrant le nombre moyen de tissus dans les lesquels les gènes humains sont exprimés, en fonction de leur première apparition dans la phylogénie. Quelque soit la source de données ou le type de jeu de données (rassemblés en familles de gènes ou séparés en caractères à chaque duplication), on peut clairement observer une diminution du nombre moyen de tissus dans lesquels les gènes sont exprimés au cours de l'évolution. Nous passons ensuite en revue les différents mécanismes qui ont sans doute menés à ce modèle : l'élargissement de l'expression des gènes au cours de l'évolution, la tendance des gènes dupliqués à se sous-fonctionnaliser, et la différenciation du nombre croissant de types de cellules et de systèmes anatomiques au cours de l'évolution. La conclusion discute de l'importance d'intégrer le contenu des génomes séquencés et les données fonctionnelles dans un cadre phylogénétique explicite tel que MANTiS ; et que malgré la faible couverture de certains génomes séquencés, de grandes imperfections dans l'annotation, et un large biais taxonomique dans la sélection des espèces séquencées, nos analyses avec MANTiS ont permis d'identifier une contrainte historique frappante dans l'expression des gènes. Nous terminons avec une section « Methods » reprenant la description du pipeline de MANTiS pour l'acquisition des données, la cartographie des caractères et les données d'expression des gènes.

TITLE AND ABSTRACT

Historical constraints on vertebrate genome evolution

Michel C. Milinkovitch[1], Raphaël Helaers[2], Athanasia C. Tzika[1,3]

Genome Biol Evol 2010: 13-18 (2010) ; doi:10.1093/gbe/evp052

[1] *Laboratory of Artificial & Natural Evolution (LANE), Dept. of Genetics & Evolution Sciences III, 30, Quai Ernest-Ansermet, 1211 Genève 4, Switzerland;*

[2] *Department of Biology & Department of Mathematics, FUNDP, rue de Bruxelles 61, 5000 Namur, Belgium*;

[3] *Evolutionary Biology & Ecology, Université Libre de Bruxelles, Av. F.D. Roosevelt 50, B-1050 Brussels, Belgium*.

Background. Recent analyses have shown that genes with larger effect of knock-out or mutation, and with larger probability to revert to single copy after whole genome duplication, are expressed earlier in development. Here, we investigate whether tissue specificity of gene expression is constrained by the age of origin of the corresponding genes.

Methodology/Principal Findings. We use 38 metazoan genomes and a comparative genomic application system to integrate inference of gene duplication with expression data from 17,503 human genes into a strictly phylogenetic framework. We show that the number of anatomical systems in which genes are expressed decreases steadily with decreased age of the genes first appearance in the phylogeny: the oldest genes are expressed, on average, in twice as many anatomical systems than the genes gained recently in evolution. These results are robust to different sources of expression data, to different levels of the anatomical system hierarchy, and to the use of gene families rather than duplication events. Finally, we show that the rate of increase in gene tissue specificity correlates with the relative rate of increase in the maximum number of cell types in the corresponding taxa.

Conclusions/Significance. Our analyses suggest that the age of first appearance of a gene in the phylogeny is highly predictive of its level of tissue specificity. Although sub-functionalization and increase in cell type number throughout evolution could constitute, respectively, the proximal and ultimate causes of this correlation, the two phenomena are intermingled. One of the biggest challenges of comparative genomics lies in the identification of changes in genome content that had significant functional implications. Our analyses identify a striking historical constraint in gene expression: the number of cell types in existence at the time of a gene appearance (through duplication or *de-novo* origination) tends to determine its level of tissue specificity for tens or hundreds of million years.

Author Summary. Understanding how genomes have changed through evolution to allow for the diversification of forms and the development of new biological functions is a formidable and challenging task. This objective is becoming feasible thanks to the recent accumulation of full genome sequences from multiple species and the development of ever more powerful computational methods. Here, we compare 38 animal genomes in a strictly phylogenetic framework and integrate reconstruction of gene duplication events with expression data from 17,503 human genes. We show that the number of anatomical systems in which genes are

expressed decreases steadily with decreased age of the genes first appearance in the phylogeny: the oldest genes are expressed, on average, in twice as many anatomical systems than the genes gained recently in animal evolution. Furthermore, this robust trend correlates with the increase in the number of cell types in animals through evolution. Tinkering with gene expression for development of new forms and functions is therefore highly restricted: the number of cell types in existence at the birth of a new gene can constrain it's level of tissue specificity for hundreds of millions years.

CONCLUSIONS

Nous pensons qu'un des plus grands défis de la génomique comparative réside dans l'identification et la cartographie sur une phylogénie robuste, des changements dans le contenu du génome qui ont une implication fonctionnelle significative. Ceci est devenu possible via l'intégration du contenu des génomes et de données fonctionnelles associées dans un même cadre phylogénétique, réalisée dans le logiciel MANTiS. Pour ce faire, nous avons rassemblé et mis en relation dans une base de données des données génomiques et des arbres de familles de protéines (depuis ENSEMBL), des données sur les processus biologiques et les fonctions moléculaires associées à ces gènes (depuis PANTHER) et des données sur l'expression de ces gènes (depuis HMDEG, GNF et eGenetics). Nous avons rassemblé les gènes orthologues en caractères, réalisé une cartographie des gains et des pertes sur une phylogénie, calculé la sur- et la sous-représentation des différentes fonctions selon cette cartographie, et isolé les caractères s'exprimant spécifiquement dans un tissu. MANTiS propose une interface conviviale permettant de manipuler graphiquement ces données, en parcourant l'arbre phylogénétique sous différentes vues (cartographie des gains et des pertes, contenu du génome, sur- et sous-représentation des différentes fonctions, distribution des gènes tissu-spécifiques). En sus de cette exploration graphique, un système de requêtes accessible à tous permet d'interroger la base de données sur des dizaines de critères combinables. Enfin, de nombreux graphiques statistiques peuvent être générés, montrant l'évolution des données au cours du temps. De plus, l'entièreté d'un jeu de données MANTiS peut être réduit au résultat d'une requête, MANTiS recalculant l'intégralité des données. L'utilisateur aura alors accès à l'ensemble des vues et des statistiques de MANTiS pour parcourir ce jeu de données réduit. Notons également que MANTiS permet d'exporter toutes les données et graphiques qu'il génère, permettant leur utilisation dans d'autres logiciels.

Nous mettons donc à disposition de la communauté un nouvel outil de génomique comparative, proposant une approche originale basée sur la phylogénie. Nous l'avons validé en reproduisant des exemples publiés, et MANTiS a prouvé qu'il permettait d'obtenir les bon résultats, et même plus, avec beaucoup plus de facilité et de précision que les techniques précédemment utilisées. Nous l'avons également comparé à d'autres bases de données, et prouvé la robustesse de sa cartographie, et l'intérêt d'utiliser des phylogénies de familles de protéines plutôt que des « Blast Reciprocal Best Hit » pour inférer les événements de duplication.

MANTiS a un grand potentiel, car il permet d'explorer des données connues, mais dans un cadre phylogénétique original. Nous l'avons prouvé au travers de 2 publications, en montrant dans un premier temps l'impact des génomes séquencés avec une faible couverture (2x) sur la cartographie des gains et des pertes, et dans un second temps qu'il existe une relation entre l'âge d'origine d'un gène et la spécificité des tissus dans lequel il s'exprime. Ces comportements peuvent déjà être observés dans les statistiques générales générées par MANTiS, l'utilisation du système de requêtes et de jeux de données réduits offrant une multitude d'autres possibilités d'analyses.

MANTiS est également ouvert et extensible, il est mis à jour régulièrement lorsqu'une nouvelle version d'ENSEMBL paraît, mais d'autres sources de données peuvent aussi y être incorporées

et bénéficier de toutes les fonctionnalités offertes par MANTiS. Par exemple, le projet Human Phylome (Huerta-Cepas et al. 2007; Huerta-Cepas et al. 2008) a été intégré comme jeu de données MANTiS, contenant une sélection d'espèces différentes d'ENSEMBL et un pipeline original pour générer les arbres de gènes. Il a par exemple permis de valider l'impact des génomes à faible couverture, car ne contenant que des espèces séquencées avec une bonne couverture. D'autres sources de données peuvent donc aisément être intégrées pour enrichir les informations fournies par MANTiS.

CONCLUSIONS ET PERSPECTIVES

Nous avons présenté MetaPIGA 2.0 et MANTiS, deux logiciels développés pour les biologistes intéressés par la phylogénie. Dans chacun d'eux, nous avons porté une attention toute particulière à la convivialité d'utilisation, en proposant des logiciels graphiques qui utilisent au maximum les standards du domaine. Nous nous sommes également efforcés de rendre exportables toutes les données qui pourraient intéresser l'utilisateur, et qu'il puisse facilement les réintroduire dans un autre logiciel. MetaPIGA rassemble de très nombreuses méthodes et outils ayant pour but commun de permettre aux méta-heuristiques implémentées d'estimer une phylogénie le plus précisément possible. Si l'intérêt de l'estimation phylogénétique est évident pour les biologistes, l'apport que peut fournir un logiciel comme MANTiS est sans doute plus difficile à estimer. Nous avons cependant prouvé au travers de trois publications qu'il s'agit d'un outil puissant ayant beaucoup de potentiel, dont les premières données ont déjà permis de mettre en évidence plusieurs comportements évolutifs d'intérêt. De nombreuses autres possibilités de recherches sont réalisables tant avec MetaPIGA qu'avec MANTiS dans leur état actuel. Et comme pour tout logiciel informatique, ils peuvent être améliorés et étendus de diverses manières, afin d'intégrer d'autres sous-domaines d'inférence phylogénétique ou de génomique comparative et s'attaquer à d'autres problématiques.

Tant MetaPIGA 2.0 que MANTiS ont donc été implémentés dans l'optique d'être étendus dans le futur. La programmation orientée-objet offerte par le langage JAVA a permis de rendre nos logiciels modulables, et nous offre donc la possibilité de développements futurs aisés. Pour conclure cette thèse, nous allons passer en revue une partie des perspectives futures de développement de MetaPIGA 2.0 et de MANTiS.

METAPIGA 2.0

Le but principal de MetaPIGA est de servir de « laboratoire » pour expérimenter de nouvelles méta-heuristiques, et inclus actuellement un Hill Climbing, un Simulated Annealing, un algorithme génétique et le metaGA. Le framework de MetaPIGA permet cependant d'ajouter facilement d'autres méta-heuristiques, à adapter au problème de l'inférence phylogénétique. L'algorithme génétique par exemple fait partie d'une famille de méta-heuristique, de type « swarm intelligence », qui est assez adaptée à l'optimisation de phylogénie et comprend d'autres méthodes telles que le « Ant Colony Optimisation », le « Stochastic Diffusion Search » ou le « Particule Swarm Optimization ». Il existe aussi d'autres approches célèbres telles que le « Tabu Search », le « GRASP » (Greedy Randomized Adaptive Search Procedure) ou le « Harmony search » (Glover and Kochenberger 2003).

Dans le même ordre d'idées, nous prévoyons également d'ajouter de nouvelles méthodes d'optimisation des longueurs de branches et des paramètres du modèle. Des algorithmes d'optimisation mathématiques spécialisés pourraient sans doute fournir de meilleures

performances que l'algorithme génétique pour un nombre restreint de paramètres (comme les méthodes DFO[34]).

Pour améliorer la convivialité et l'aide à la décision, nous pensons adjoindre à MetaPIGA un paramétrage automatique qui utiliserait une série d'outils déjà implémentés (choix du modèle le plus adéquat, optimisation de ses paramètres, arrêt automatique de l'heuristique et du nombre de réplicats), mais qui se baserait également sur les caractéristiques du jeu de données (nombre de séquences, nombre de sites, divergence des séquences). En testant différents ensembles de paramètres pour une série de jeux de données représentatifs, nous aurions une base de données nous permettant de choisir l'heuristique la plus adéquate avec son paramétrage pour un jeu de donnée ayant des caractéristiques similaires. Une petite série de questions simples posées à l'utilisateur (la vitesse à laquelle il veut un résultat, son degré de précision, etc.) permettrait d'affiner encore plus ce paramétrage automatique.

Un travail conséquent pourra être effectué avec la version actuelle de MetaPIGA 2.0 : (1) la recherche des paramètres les plus adaptés à chaque heuristique et méta-heuristique, (2) la comparaison de l'efficacité des différentes méta-heuristiques implémentées, et (3) la comparaison de performances entre la meilleure méta-heuristique implémentée dans MetaPIGA et celles implémentées dans d'autres logiciels (PhyML, MrBayes, etc).

MetaPIGA ne peut actuellement traiter que des données nucléotidiques. Dans sa conception actuelle, il pourrait facilement traiter des jeux de données protéiques, par exemple avec un modèle GTR similaire aux substitutions de nucléotides ou des matrices BLOSUM (Henikoff & Henikoff 1992). Nous envisageons donc de permettre à MetaPIGA de prendre en charge non seulement des jeux de données protéiques, mais d'aller plus loin en généralisant le principe à n'importe quel nombre d'états. Nous pourrions par exemple avoir un jeu de données à 2 états (0/1, présence/absence d'un caractère par exemple) et le gérer avec un modèle GTR ajusté à 2 états.

Enfin, nous envisageons d'offrir à l'utilisateur la possibilité d'optimiser les éléments de la matrice de taux (relatifs instantanés) séparément pour chaque branche de l'arbre, mais également de lui permettre de générer les séquences ancestrales de son jeu de données en se basant sur les valeurs de vraisemblance conditionnelle.

MANTIS

L'extension la plus simple pour MANTiS, et ne nécessitant aucune modification de son code, est l'ajout de nouveaux jeux de données, autres que ceux liés à ENSEMBL. Un jeu de données basé sur le projet « Human Phylome » (Huerta-Cepas *et al.* 2007) a déjà été ajouté, et nous prévoyons d'intégrer un jeu de données basé sur des espèces végétales.

Vu le grand nombre de comparaisons faites pour tester la significativité de la sur/sous représentation des données fonctionnelles, la chance d'échantillonner des faux positifs

[34] **DFO** : Derivate-Free Optimization. Englobe des méthodes d'optimisation mathématiques qui ne nécessitent pas de connaître les dérivées de la fonction objectif.

augmente avec le nombre de test. Afin de pallier à cela , une correction de multi-testing devrait être implémentée dans la prochaine version de MANTiS, par exemple en appliquant une correction de Bonferonni sur les p-values calculées.

Au niveau des données d'expression, nous sommes à la recherche d'une source de données qui serait plus complète et plus cohérente que les données GNF et eGenetics reprises dans la base de données ENSEMBL, qui contiennent énormément de gènes classés dans la catégorie « no information ». De plus, MANTiS propose de comparer les gènes tissu-spécifiques entre GNF, eGenetics et HMDEG, et une simple observation des résultats montre que les 3 bases de données ne sont pas toujours en accord (par exemple, pour l'une un gène donné est exprimé spécifiquement dans le système nerveux, alors que pour l'autre il est exprimé spécifiquement dans le système alimentaire). Nous pensons donc à terme remplacer GNF et eGenetics par une nouvelle source de données d'expression, que nous comparerons à HMDEG qui semble être la plus complète. Une autre possibilité serait d'analyser directement les données brutes de ces bases de données. Ces 2 bases de données n'ayant que des données sur l'être humain, contrairement à HMDEG qui propose également des données sur la souris, les remplacer permettrait peut-être également d'étendre les données d'expression de MANTiS à la souris (voir à d'autres espèces).

Nous prévoyons également d'inclure de nouveaux types de données dans un futur proche, ainsi qu'une nouvelle vue MANTiS associée. Il s'agit par exemple des réseaux d'interaction protéine-protéine, ou de données de microARNs.

Nous développons actuellement une vue permettant d'interagir graphiquement et de manière dynamique avec un réseau d'interactions. Cette vue permettra de manipuler un réseau d'interaction protéine-protéine, et d'afficher aisément des sous-réseaux en sélectionnant les gènes à la main, où en les filtrant grâce à divers critères propres à MANTiS (comme les fonctions associées aux gènes ou sa cartographie sur la phylogénie). Une fois que le réseau désiré sera affiché, des outils annexes permettront de visualiser graphiquement le comportement du réseau. Par exemple, il sera possible de donner aux nœuds une taille proportionnelle à leur degré (c'est-à-dire le nombre de voisin directs), de griser une partie du réseau selon certains critères (gènes associés à certains processus biologiques ou fonctions moléculaire, exprimé dans certains tissus, exprimés spécifiquement, gènes perdu dans un certain nombre de branches), d'afficher le « plus court chemin » entre 2 nœuds du réseau ou de colorer les voisons des nœuds sélectionné selon leur niveau de relation (voisins directs, voisins des voisins, etc). Un algorithme de clustering permettra d'éliminer (griser) un nombre donné d'arcs, qui seront choisi sur base de la densité de connexion des clusters obtenus. Cliquer sur un nœud permettra d'obtenir la totalité des informations détenue par MANTiS sur le gène associé (description, fonctions associées, événements de duplications, orthologues, etc). Enfin, une nouvelles séries de statistiques globales sur l'interactome complet seront générées, et il sera possible d'obtenir toute une série d'histogrammes et de mesures descriptives (moyenne, variance, écart-type, etc.) sur le sous-réseau affiché par l'utilisateur (telles que la distribution du degré, du diamètre du réseau, du nombre de pertes, du coefficient de clustering ou de la taille des clusters). L'interactomique comparative (la comparaison de réseaux d'interaction protéine-protéine parmi plusieurs espèces) en étant à ses débuts, principalement à cause du peu de recouvrement parmi les jeux de données de

différentes espèces, MANTiS devrait permettre de faciliter l'analyse de l'évolubilité et de la robustesse des réseaux d'interaction protéine-protéine des eukaryotes dans un cadre phylogénétique.

Un autre type de données qui pourrait être intégré à MANTiS sont les microARNs. Les microARNs (miRNAs) forment une classe de gènes ARN non-codant dont les produits sont de petits ARN simple brin d'une longueur d'environ 22 nucléotides. Ils sont impliqués dans la régulation de la traduction et de la dégradation de l'ARN messager. Plus de 10.000 microARNs provenant de diverses espèces animales et végétales ont été reportées jusqu'ici, et une base de données dédiée, miRBASE (Griffiths-Jones *et al.* 2008), a été créée pour rassembler les informations à leur sujet. Peu de choses sont connues sur l'histoire évolutive de ces éléments régulateurs et encore moins sur leurs cibles potentielles. Le cadre phylogénétique de MANTiS pourrait être ajusté à ces jeux de données additionnels : d'une part, les microARNs eux-mêmes et d'autre part, leurs séquences cibles. Cette approche nous mènerait non seulement à l'identification de nouvelles paires microARN/cible, mais ferait également la lumière sur la manière dont ces séquences évoluent.

RÉFÉRENCES

GÉNÉRALES

Alexeyenko, A., Tamas, I., Liu, G., and Sonnhammer, E.L. (2006). "Automatic clustering of orthologs and inparalogs shared by multiple proteomes". *Bioinformatics (Oxford, England)* **22**(14):e9-15.

Bashir, A., Ye, C., Price, A.L. and Bafna, V. (2005) "Orthologous repeats and mammalian phylogenetic inference". *Genome Res. %R 10.1101/gr.3493405*, **15**, 998-1006.

Blair, J.E., Ikeo, K., Gojobori, T., and Hedges, B. (2002). "The evolutionary position of nematodes". *BMC Evolutionary Biology* **2**:7.

Box, G.E.P., Hunter, W.G., and Hunter, J.S. (1978). "Statistics for experimenters". *Wiley,* p53.

Camin, J.H. and Sokal, R.R. (1965). "A Method for Deducing Branching Sequences in Phylogeny". *Evolution* **19**(3):311-326.

De Jong, K. (1988). "Learning with genetic algorithms : An overview". *Machine Learning* **3**(2):121-138.

DiDonato, A.R. and Morris, A.H. (1986). "Computation of the incomplete gamma function ratios and their inverse". *ACM Transactions on Mathematical Software* **12**(4):377-393.

Dubchak, I. and Frazer, K. (2003). "Multi-species sequence comparison: the next frontier in genome annotation". *Genome Biology* **4**:122.

Felsenstein, J. (2004). "Inferring phylogenies". *Sinauer Associates, Inc.*

Felsenstein, J. (2005). "Phylip (phylogeny inference package) version 3.6". Distributed by the author.

Forster, M.R., Sober, E. (2004) "Why likelihood?". In: *Likelihood and Evidence* (eds. Taper M, Lele S). University of Chicago Press, Chicago.

Gabaldon, T. (2008). "Large-scale assignment of orthology: back to phylogenetics?". *Genome biology* **9**(10):235.

Glover, F. and Kochenberger, G.A. (2003). "Handbook of metaheuristics". *Kluwer Academic Publishers*.

Griffiths-Jones, S., Saini, H.K., van Dongen, S., and Enright, A.J. (2008). "miRBase: tools for microRNA genomics". *Nucleic Acids Res*, **36**, D154-158.

Gu, X., Fu, Y.-X., and Li, W.-H. (1995). "Maximum likelihood estimation of the heterogeneity of substitution rate among nucleotide sites". *Mol. Biol. Evol.* **12**: 546-557.

Guindon, S. and Gascuel, O. (2003). “A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood”. *Syst Biol*, **52**(5), 696-704.

Halanych, K.M. (2004). “The New View of Animal Phylogeny”, *Annual Review of Ecology, Evolution, and Systematics*, **35**, 229-256.

Hasegawa M., Kishino, H., and Yano, T. (1985). “Dating the human-ape splitting by a molecular clock of mitochondrial DNA”. *Journal of Molecular Evolution* **22**:160-174.

Henikoff, S. and Henikoff (1992). "Amino acid substitution matrices from protein blocks". *Proceedings of the National Academy of Sciences of the United States of America* **89** (22): 10915–9.

Hillis, D. M., Moritz, C., and Mable, B. K. (1996). “Molecular Systematics”. *Sinauer Associates, Inc*.

Hubbard, T.J., Aken, B.L., Beal, K., Ballester, B., Caccamo, M., Chen, Y., Clarke, L., Coates, G., Cunningham, F., Cutts, T., Down, T., Dyer, S.C., Fitzgerald, S., Fernandez-Banet, J., Graf, S., Haider, S., Hammond, M., Herrero, J., Holland, R., Howe, K., Howe, K., Johnson, N., Kahari, A., Keefe, D., Kokocinski, F., Kulesha, E., Lawson, D., Longden, I., Melsopp, C., Megy, K., Meidl, P., Ouverdin, B., Parker, A., Prlic, A., Rice, S., Rios, D., Schuster, M., Sealy, I., Severin, J., Slater, G., Smedley, D., Spudich, G., Trevanion, S., Vilella, A., Vogel, J., White, S., Wood, M., Cox, T., Curwen, V., Durbin, R., Fernandez-Suarez, X.M., Flicek, P., Kasprzyk, A., Proctor, G., Searle, S., Smith, J., Ureta-Vidal, A. and Birney, E. (2007) Ensembl 2007, *Nucleic Acids Res*, **35**, D610-617.

Huerta-Cepas, J., H. Dopazo, J. Dopazo, and T. Gabaldon. 2007. The human phylome. Genome Biol **8**:R109.

Huelsenbeck, J.P., Ronquist, F. (2001). “Mr Bayes: Bayesian inference of phylogenetic trees”. *Bioinformatics*, **17**(8), 754-755.

Jukes, TH., and Cantor, CR. (1969). “Evolution of protein molecules”. Pp. 21-123 in H. N. Munro, ed. *Mammalian protein metabolism*. Academic Press, New York.

Kass R.E., Wasserman L. (1995). “A reference Bayesian test for nested hypotheses and its relationship to the Schwarz criterion”. *Journal of the American Statistical Association* **90**, 928-934.

Kelso, J., Visagie, J., Theiler, G., Christoffels, A., Bardien, S., Smedley, D., Otgaar, D., Greyling, G., Jongeneel, C.V., McCarthy, M.I., Hide, T. and Hide, W. (2003) eVOC: a controlled vocabulary for unifying gene expression data, *Genome Res*, **13**, 1222-1230.

Kimura, M. (1968). “Evolutionary rate at the molecular level”. *Nature* **217**(5129):624-626.

Kimura, M. (1980). “A simple method for estimating evolutionary rate of base substitution through comparative studies of nucleotide sequences”. *Journal of Molecular Evolution* **16**:111-120.

Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. (1983). “Optimization by simulated annealing”. Science, **220**(4598), 671-680.

Kruger, A., Hofmann, O., Carninci, P., Hayashizaki, Y., and Hide, W. (2007). "Simplified ontologies allowing comparison of developmental mammalian gene expression". *GENOME BIOLOGY* **8**:R229.

Kullback, S., Leibler, R.A. (1951). "On information and sufficiency". *Annals of Mathematical Statistics* **22**, 79-86.

Kumar, S., Nei, M., Dudley, J., and Tamura, K. (2008). "Mega : A biologist-centric software for evolutionary analysis of dna and protein sequences". *Briefings in bioinformatics*.

Lemmon, A.R., and Milinkovitch, M.C. (2002). "The metapopulation genetic algorithm : An efficient solution for the problem of large phylogeny estimation". *Proc Natl Acad Sci USA*, **99**(16), 10516-10521.

Lewis, P.O. (1998). "A genetic algorithm for maximum-likelihood phylogeny inference using nucleotide sequence data". *Mol Biol Evol*, **15**(3), 277-283.

Li, L., Stoeckert, C.J., Jr., and Roos, D.S. (2003) "OrthoMCL: identification of ortholog groups for eukaryotic genomes". Genome Res 13: 2178-2189.

Li, W.-H. (1999). "Molecular Evolution". *Sinauer, Sunderland, MA*.

Liolios, K., Tavernarakis, N., Hugenholtz, P. and Kyrpides, N.C. (2006). "The Genomes On Line Database (GOLD) v.2: a monitor of genome projects worldwide", *Nucleic Acids Res*, **34**, D332-334.

Lundy, M. (1985). "Applications of the annealing algorithm to combinatorial problems in statistics". *Biometrika*, **72**(1), 191-198.

Lynch, M. and Conery, J.S. (2000). "The evolutionary fate and consequences of duplicate genes". *Sciences* **10**(290):1151-1155.

Maddison, D.R., Swofford, D.L., and Maddison, W.P. (1997). "NEXUS: an extensible file format for systematic information". *Syst Biol* **46**(4):590-621.

Mi, H., Guo, N., Kejariwal, A. and Thomas, P.D. (2007) PANTHER version 6: protein sequence and function evolution data with expanded representation of biological pathways, *Nucleic Acids Res*, **35**, D247-252.

Michener, C., and Sokal, R. (1957). "A quantitative approach to a problem in classification". *Evolution* **11**:130-162.

Mitchel, M. and Holland, J.H. (2003). "When will a genetic algorithm outperform hill climbing ?". In *Proceedings of the 5th International Conference on Genetic Algorithms*. Morgan Kaufmann, USA.

NIST/SEMATECH, '6.3.3.1. Counts Control Charts', *e-Handbook of Statistical Methods*, <http://www.itl.nist.gov/div898/handbook/pmc/section3/pmc331.htm> [accessed 25 October 2009]

Ohno, S. (1970). “Evolution by Gene Duplication”. *Springer-Verlag, Heidelberg, Germany*.

Pao, S.Y., Lin, W.L. and Hwang, M.J. (2006) In silico identification and comparative analysis of differentially expressed genes in human and mouse tissues, *BMC Genomics*, **7**, 86.

Posada, D. and Crandall, K. (1998). “Modeltest: testing the model of dna substitution”. *Bioinformatics*, **14**(9), 817-818.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P. (2002). “Numerical recipes in C++”, *Cambridge University Press*.

Raes, J. and Van de Peer, Y. (2003). “Gene duplication, the evolution of novel gene functions, and detecting functional divergence of duplicates in silico”. *Applied Bionformatics*, **2**(2):91-101.

Reinholtz, K. (2000). “Java will be faster than C++”. *ACM SIGPLAN Notices* **35**(2):25-28.

Ronquist, F. and Huelsenbeck, J.P. (2003). “MrBayes 3: Bayesian phylogenetic inference under mixed models”. *Bioinformatics*, **19**(12), 1572-1574.

Rzhetsky, A., and Nei, M. (1993). “Theoretical foundation of the minimum-evolution method of phylogenetic inference”. *Molecular Biology and Evolution* **10**:1073-1095.

Saitou, N., and Nei, M. (1987). "The neighbor-joining method: a new method for reconstructing phylogenetic trees". *Mol Biol Evol* **4** (4): 406-425.

Salter, L.A. and Pearl, D.K. (2001). “Stochastic search strategy for estimation of maximum likelihood phylogenetic trees”. *Systematic Biology*.

Schadt, E.E., Sinsheimer, J.S., and Lange, K. (1998). ” Computational Advances in Maximum Likelihood Methods for Molecular Phylogeny”.*Genome Res.* **8**:222-233.

Schmidt, H.A., Strimmer, K., Vingron, M., and von Haeseler, A. (2002). “Tree-Puzzle: maximum likelihood phylogenetic analysis using quartets and parallel computing”. *Bioinformatics*, **18**(3), 502-504.

Simon, D. and Larget, B. (2000). “Bayesian analysis in molecular biology and evolution (bambe)”.

Springer, M.S., Stanhope, M.J., Madsen, O. and de Jong, W.W. (2004). “Molecules consolidate the placental mammal tree”, *Trends in ecology & evolution (Personal edition)*, **19**, 430-438.

Su, A.I., Cooke, M.P., Ching, K.A., Hakak, Y., Walker, J.R., Wiltshire, T., Orth, A.P., Vega, R.G., Sapinoso, L.M., Moqrich, A., Patapoutian, A., Hampton, G.M., Schultz, P.G. and Hogenesch, J.B. (2002) Large-scale analysis of the human and mouse transcriptomes, *Proc Natl Acad Sci U S A*, **99**, 4465-4470.

Swofford, D.L. (2003). “PAUP*. Phylogenetic analysis using parsimony (*and other methods) version 4”. Sinauer Associates, Sunderland, Massachusetts.

Tamura, K., and Nei, M. (1993). “Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees”. *Molecular Biology and Evolution* **10**:512-526.

Tavaré, S. (1986). "Some Probabilistic and Statistical Problems in the Analysis of DNA Sequences". *American Mathematical Society: Lectures on Mathematics in the Life Sciences* **17**: 57–86 (1986).

Vilella, A.J., Severin, J., Ureta-Vidal, A., Heng, L., Durbin, R., and Birney, E. (2009). “EnsemblCompara GeneTrees: Complete, duplication-aware phylogenetic trees in vertebrates”. *Genome Res* **19**(2):327-335.

Waddell, P.J., and Steel, M.A. (1997). “General Time-Reversible Distances with Unequal Rates across Sites: Mixing Gamma and Inverse Gaussian Distributions with Invariant Sites”. *Molecular Phylogenetics and Evolution* Vol. 3, No 3, December, pp. 398-414.

Waddell, P.J., and Penny, D. (1996). “Evolutionary trees of apes and humans from DNA sequence”. In “*Handbook of Human Symbolic Evolution*” (A. J. Lock and C.R. Peters, Eds.), pp. 53-73, Oxford Univ. Press, Oxford.

Wasserman, L. (2000). “Bayesian Model Selection and Model Averaging”. *J Math Psychol* **44**, 92-107.

Wen-Hsiung, L. (1997). “Molecular Evolution”. *Sinauer Associates, Inc*.

Woese, C. R. (1998). “The Universal Ancestor”. *Proceedings of the National Academy of Sciences* **95**: 6854-6859.

Xia, X. and Xie, Z. (2001). “Dambe: Software package for data analysis in molecular biology and evolution”. *J Hered*, **92**(4), 371-373.

Yang, Z. (1994). “Maximum Likelihood Phylogenetic Estimation from DNA Sequences with Variable Rates over Sites: Approximate Methods”. *J Mol Evol* (1994) **39**:306-314.

Zwickl, D.J. (2006). “Genetic algorithm approaches for the phylogenetic analysis of large biological sequence datasets under the maximum likelihood criterion”. Ph.D. thesis, The University of Texas at Austin.

METAPIGA 2.0 : MAXIMUM LIKELIHOOD LARGE PHYLOGENY ESTIMATION USING THE METAPOPULATION GENETIC ALGORITHM AND OTHER STOCHASTIC HEURISTICS

Gabaldon T: **Large-scale assignment of orthology: back to phylogenetics?** *Genome Biol* 2008, **9**(10):235.

Li W-H: **Molecular evolution.** Sunderland, MA.: Sinauer; 1997.

Thorne JL, Kishino H: **Divergence time and evolutionary rate estimation with multilocus data**. *Syst Biol* 2002, **51**(5):689-702.

Cassens I, Vicario S, Waddell VG, Balchowsky H, Van Belle D, Ding W, Fan C, Mohan RS, Simoes-Lopes PC, Bastida R *et al*: **Independent adaptation to riverine habitats allowed survival of ancient cetacean lineages**. *Proc Natl Acad Sci U S A* 2000, **97**(21):11343-11347.

Thorne JL, Kishino H, Painter IS: **Estimating the rate of evolution of the rate of molecular evolution**. *Molecular Biology and Evolution* 1998, **15**(12):1647-1657.

Blanchette M, Green ED, Miller W, Haussler D: **Reconstructing large regions of an ancestral mammalian genome in silico**. *Genome Res* 2004, **14**(12):2412-2423.

Chang BS, Jonsson K, Kazmi MA, Donoghue MJ, Sakmar TP: **Recreating a functional ancestral archosaur visual pigment**. *Molecular biology and evolution* 2002, **19**(9):1483-1489.

Chang BS, Ugalde JA, Matz MV: **Applications of ancestral protein reconstruction in understanding protein function: GFP-like proteins**. *Methods Enzymol* 2005, **395**:652-670.

Williams PD, Pollock DD, Blackburne BP, Goldstein RA: **Assessing the accuracy of ancestral protein reconstruction methods**. *PLoS computational biology* 2006, **2**(6):e69.

Zhang J, Nielsen R, Yang Z: **Evaluation of an improved branch-site likelihood method for detecting positive selection at the molecular level**. *Molecular biology and evolution* 2005, **22**(12):2472-2479.

Meegaskumbura M, Bossuyt F, Pethiyagoda R, Manamendra-Arachchi K, Bahir M, Milinkovitch MC, Schneider CJ: **Sri Lanka: an amphibian hot spot**. *Science* 2002, **298**(5592):379.

Springer MS, Stanhope MJ, Madsen O, de Jong WW: **Molecules consolidate the placental mammal tree**. *Trends in ecology & evolution (Personal edition)* 2004, **19**(8):430-438.

Bossuyt F, Brown RM, Hillis DM, Cannatella DC, Milinkovitch MC: **Phylogeny and biogeography of a cosmopolitan frog radiation: Late cretaceous diversification resulted in continent-scale endemism in the family ranidae**. *Syst Biol* 2006, **55**(4):579-594.

Graham RL, Foulds LR: **Unlikelihood that Minimal Phylogenies for a Realistic Biological Study Can Be Constructed in Reasonable Computational Time**. *Math Bioscience* 1982, **60**:133-142.

Chor B, Tuller T: **Maximum likelihood of evolutionary trees: hardness and approximation**. *Bioinformatics* 2005, **21 Suppl 1**:i97-106.

Felsenstein J: **Evolutionary trees from DNA sequences: a maximum likelihood approach**. *Journal of molecular evolution* 1981, **17**:368-376.

Holder M, Lewis PO: **Phylogeny estimation: traditional and Bayesian approaches**. *Nat Rev Genet* 2003, **4**(4):275-284.

Huelsenbeck JP, Larget B, Miller RE, Ronquist F: **Potential applications and pitfalls of Bayesian inference of phylogeny**. *Syst Biol* 2002, **51**(5):673-688.

Salter LA, Pearl DK: **Stochastic search strategy for estimation of maximum likelihood phylogenetic trees**. *Syst Biol* 2001, **50**(1):7-17.

Katoh K, Kuma K, Miyata T: **Genetic algorithm-based maximum-likelihood analysis for molecular phylogeny**. *J Mol Evol* 2001, **53**(4-5):477-484.

Lemmon AR, Milinkovitch MC: **The metapopulation genetic algorithm: An efficient solution for the problem of large phylogeny estimation**. *Proc Natl Acad Sci U S A* 2002, **99**(16):10516-10521.

Lewis PO: **A genetic algorithm for maximum-likelihood phylogeny inference using nucleotide sequence data**. *Mol biol evol* 1998, **15**(3):277-283.

Matsuda H: **Protein phylogenetic inference using maximum likelihood with a genetic algorithm**. In: *Pacific symposium on biocomputing '96: 1996; London*: World Scientific; 1996: 512-523.

Zwickl DJ: **Genetic algorithm approaches for the phylogenetic analysis of large biological sequence datasets under the maximum likelihood criterion**. Austin, Tx, USA.: The University of Texas; 2006.

Ronquist F, Huelsenbeck JP: **MrBayes 3: Bayesian phylogenetic inference under mixed models**. *Bioinformatics* 2003, **19**(12):1572-1574.

Stamatakis A: **RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models**. *Bioinformatics* 2006, **22**(21):2688-2690.

Suchard MA, Rambaut A: **Many-core algorithms for statistical phylogenetics**. *Bioinformatics* 2009, **25**(11):1370-1376.

Tavaré S: **Some Probabilistic and Statistical Problems in the Analysis of DNA Sequences**. *American Mathematical Society: Lectures on Mathematics in the Life Sciences* 1986, **17**:57–86.

Yang Z: **Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods**. *J Mol Evol* 1994, **39**(3):306-314.

Yang Z: **Among-site rate variation and its impact on phylogenetic analyses**. *Trends in Ecology & Evolution* 1996, **11**(9):367-372.

Gu X, Fu YX, Li WH: **Maximum likelihood estimation of the heterogeneity of substitution rate among nucleotide sites**. *Mol biol evol* 1995, **12**(4):546-557.

Guindon S, Gascuel O: **A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood**. *Syst Biol* 2003, **52**(5):696-704.

Stamatakis A, Ludwig T, Meier H: **RAxML-III: a fast program for maximum likelihood-based inference of large phylogenetic trees**. *Bioinformatics* 2005, **21**(4):456-463.

Saitou N, Nei M: **The neighbor-joining method: a new method for reconstructing phylogenetic trees**. *Molecular Biology and Evolution* 1987, **4**(4):406-425.

Felsenstein J: **Inferring Phylogenies**. Sunderland: Sinauer Associates Inc.; 2002.

Kirkpatrick S, Gelatt CD, Jr., Vecchi MP: **Optimization by Simulated Annealing**. *Science* 1983, **220**(4598):671-680.

Lundy M: **Applications of the Annealing Algorithm to Combinatorial Problems in Statistics**. *Biometrika* 1985, **72**(1):191-198.

Holland J: **Adaptation in Natural and Artificial Systems**. Ann Arbor: University of Michigan Press; 1975.

Maddison DR, Swofford DL, Maddison WP: **NEXUS: an extensible file format for systematic information**. *Syst Biol* 1997, **46**(4):590-621.

Posada D, Crandall KA: **Selecting the best-fit model of nucleotide substitution**. *Syst Biol* 2001, **50**(4):580-601.

MANTIS: A PHYLOGENETIC FRAMEWORK FOR MULTI-SPECIES GENOME COMPARISONS

Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M. and Sherlock, G. (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium, *Nat Genet*, **25**, 25-29.

Belov, K., Sanderson, C.E., Deakin, J.E., Wong, E.S., Assange, D., McColl, K.A., Gout, A., de Bono, B., Barrow, A.D., Speed, T.P., Trowsdale, J. and Papenfuss, A.T. (2007) Characterization of the opossum immune genome provides insights into the evolution of the mammalian immune system, *Genome Res*.

Bray, N., Dubchak, I. and Pachter, L. (2003) AVID: A global alignment program, *Genome Res*, **13**, 97-102.

Brudno, M., Poliakov, A., Minovitsky, S., Ratnere, I. and Dubchak, I. (2007) Multiple whole genome alignments and novel biomedical applications at the VISTA portal, *Nucleic Acids Res*.

Curwen, V., Eyras, E., Andrews, T.D., Clarke, L., Mongin, E., Searle, S.M. and Clamp, M. (2004) The Ensembl automatic gene annotation system, *Genome Res*, **14**, 942-950.

Dufayard, J.F., Duret, L., Penel, S., Gouy, M., Rechenmann, F. and Perriere, G. (2005) Tree pattern matching in phylogenetic trees: automatic search for orthologs or paralogs in homologous gene sequence databases, *Bioinformatics*, **21**, 2596-2603.

Edgar, R.C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput, *Nucleic Acids Res*, **32**, 1792-1797.

Fleischmann, R.D., Adams, M.D., White, O., Clayton, R.A., Kirkness, E.F., Kerlavage, A.R., Bult, C.J., Tomb, J.F., Dougherty, B.A., Merrick, J.M. and et al. (1995) Whole-genome random sequencing and assembly of Haemophilus influenzae Rd, *Science*, **269**, 496-512.

Force, A., Lynch, M., Pickett, F.B., Amores, A., Yan, Y.L. and Postlethwait, J. (1999) Preservation of duplicate genes by complementary, degenerative mutations, *Genetics*, **151**, 1531-1545.

Gish, W. (1996-2004).

Gouret, P., Vitiello, V., Balandraud, N., Gilles, A., Pontarotti, P. and Danchin, E.G. (2005) FIGENIX: intelligent automation of genomic annotation: expertise integration in a new software platform, *BMC bioinformatics*, **6**, 198.

Greer, J.M., Puetz, J., Thomas, K.R. and Capecchi, M.R. (2000) Maintenance of functional equivalence during paralogous Hox gene evolution, *Nature*, **403**, 661-665.

Guindon, S. and Gascuel, O. (2003) A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood, *Syst Biol*, **52**, 696-704.

He, X. and Zhang, J. (2005) Rapid subfunctionalization accompanied by prolonged and substantial neofunctionalization in duplicate gene evolution, *Genetics*, **169**, 1157-1164.

Hubbard, T.J., Aken, B.L., Beal, K., Ballester, B., Caccamo, M., Chen, Y., Clarke, L., Coates, G., Cunningham, F., Cutts, T., Down, T., Dyer, S.C., Fitzgerald, S., Fernandez-Banet, J., Graf, S., Haider, S., Hammond, M., Herrero, J., Holland, R., Howe, K., Howe, K., Johnson, N., Kahari, A., Keefe, D., Kokocinski, F., Kulesha, E., Lawson, D., Longden, I., Melsopp, C., Megy, K., Meidl, P., Ouverdin, B., Parker, A., Prlic, A., Rice, S., Rios, D., Schuster, M., Sealy, I., Severin, J., Slater, G., Smedley, D., Spudich, G., Trevanion, S., Vilella, A., Vogel, J., White, S., Wood, M., Cox, T., Curwen, V., Durbin, R., Fernandez-Suarez, X.M., Flicek, P., Kasprzyk, A., Proctor, G., Searle, S., Smith, J., Ureta-Vidal, A. and Birney, E. (2007) Ensembl 2007, *Nucleic Acids Res*, **35**, D610-617.

Hurles, M. (2004) Gene duplication: the genomic trade in spare parts, *PLoS Biol*, **2**, E206.

Kelso, J., Visagie, J., Theiler, G., Christoffels, A., Bardien, S., Smedley, D., Otgaar, D., Greyling, G., Jongeneel, C.V., McCarthy, M.I., Hide, T. and Hide, W. (2003) eVOC: a controlled vocabulary for unifying gene expression data, *Genome Res*, **13**, 1222-1230.

Kidd, K.K. and Sgaramella-Zonta, L.A. (1971) Phylogenetic analysis: concepts and methods, *Am J Hum Genet*, **23**, 235-252.

Kondrashov, F.A. and Kondrashov, A.S. (2006) Role of selection in fixation of gene duplications, *Journal of theoretical biology*, **239**, 141-151.

Liolios, K., Tavernarakis, N., Hugenholtz, P. and Kyrpides, N.C. (2006) The Genomes On Line Database (GOLD) v.2: a monitor of genome projects worldwide, *Nucleic Acids Res*, **34**, D332-334.

Lynch, M. and Conery, J.S. (2000) The evolutionary fate and consequences of duplicate genes, *Science*, **290**, 1151-1155.

Lynch, M. and Force, A. (2000) The probability of duplicate gene preservation by subfunctionalization, *Genetics*, **154**, 459-473.

Mi, H., Guo, N., Kejariwal, A. and Thomas, P.D. (2007) PANTHER version 6: protein sequence and function evolution data with expanded representation of biological pathways, *Nucleic Acids Res*, **35**, D247-252.

Odronitz, F., Hellkamp, M. and Kollmar, M. (2007) diArk--a resource for eukaryotic genome research, *BMC Genomics*, **8**, 103.

Ohno, S. (1970) *Evolution by gene duplication*. Springler Verlag, Heidelberg.

Pao, S.Y., Lin, W.L. and Hwang, M.J. (2006) In silico identification and comparative analysis of differentially expressed genes in human and mouse tissues, *BMC Genomics*, **7**, 86.

Rastogi, S. and Liberles, D.A. (2005) Subfunctionalization of duplicated genes as a transition state to neofunctionalization, *BMC evolutionary biology*, **5**, 28.

Rzhetsky, A. and Nei, M. (1992) Statistical properties of the ordinary least-squares, generalized least-squares, and minimum-evolution methods of phylogenetic inference, *Journal of molecular evolution*, **35**, 367-375.

Rzhetsky, A. and Nei, M. (1993) Theoretical foundation of the minimum-evolution method of phylogenetic inference, *Molecular biology and evolution*, **10**, 1073-1095.

Shiu, S.H., Byrnes, J.K., Pan, R., Zhang, P. and Li, W.H. (2006) Role of positive selection in the retention of duplicate genes in mammalian genomes, *Proc Natl Acad Sci U S A*, **103**, 2232-2236.

Stalker, J., Gibbins, B., Meidl, P., Smith, J., Spooner, W., Hotz, H.R. and Cox, A.V. (2004) The Ensembl Web site: mechanics of a genome browser, *Genome Res*, **14**, 951-955.

Su, A.I., Cooke, M.P., Ching, K.A., Hakak, Y., Walker, J.R., Wiltshire, T., Orth, A.P., Vega, R.G., Sapinoso, L.M., Moqrich, A., Patapoutian, A., Hampton, G.M., Schultz, P.G. and Hogenesch, J.B. (2002) Large-scale analysis of the human and mouse transcriptomes, *Proc Natl Acad Sci U S A*, **99**, 4465-4470.

Zhang, J. (2003) Evolution by gene duplication: an update, *Trends in Ecology and Evolution*, **18**, 292-298.

MAPPING GENE GAINS AND LOSSES AMONG METAZOAN FULL GENOMES USING AN INTEGRATED PHYLOGENETIC FRAMEWORK

Aburomia, R., O. Khaner, and A. Sidow. 2003. Functional evolution in the ancestral lineage of vertebrates or when genomic complexity was wagging its morphological tail Journal of Structural and Functional Genomics:45-52.

Alexeyenko, A., I. Tamas, G. Liu, and E. L. Sonnhammer. 2006. Automatic clustering of orthologs and inparalogs shared by multiple proteomes. Bioinformatics **22**:e9-15.

Bashir, A., C. Ye, A. L. Price, and V. Bafna. 2005. Orthologous repeats and mammalian phylogenetic inference. Genome Res. **15**:998-1006.

Bejerano, G., M. Pheasant, I. Makunin, S. Stephen, W. J. Kent, J. S. Mattick, and D. Haussler. 2004. Ultraconserved Elements in the Human Genome. Science **304**:1321-1325.

Carroll, S., J. Grenier, and S. Weatherbee. 2004. From DNA to Diversity: Molecular Genetics and the Evolution of Animal Design. Wiley-Blackwell.

Carroll, S. B. 2005. Evolution at two levels: On genes and form. Plos Biology **3**:1159-1166.

Carroll, S. B. 2001. Chance and necessity: the evolution of morphological complexity and diversity. Nature **409**:1102-1109.

Carroll, S. B., J. K. Grenier, and S. D. Weatherbee. 2005. From DNA to diversity, molecular genetics and the evolution of animal design. Blackwell publishing, Malden.

Clark, A. G., S. Glanowski, R. Nielsen, P. D. Thomas, A. Kejariwal, M. A. Todd, D. M. Tanenbaum, D. Civello, F. Lu, B. Murphy, S. Ferriera, G. Wang, X. Zheng, T. J. White, J. J. Sninsky, M. D. Adams, and M. Cargill. 2003. Inferring Nonneutral Evolution from Human-Chimp-Mouse Orthologous Gene Trios. Science **302**:1960-1963.

Deluca, T. F., I. H. Wu, J. Pu, T. Monaghan, L. Peshkin, S. Singh, and D. P. Wall. 2006. Roundup: a multi-genome repository of orthologs and evolutionary distances. Bioinformatics **22**:2044-2046.

Dennis, G., Jr., B. T. Sherman, D. A. Hosack, J. Yang, W. Gao, H. C. Lane, and R. A. Lempicki. 2003. DAVID: Database for Annotation, Visualization, and Integrated Discovery. Genome Biol **4**:P3.

Dermitzakis, E. T., A. Reymond, N. Scamuffa, C. Ucla, E. Kirkness, C. Rossier, and S. E. Antonarakis. 2003. Evolutionary Discrimination of Mammalian Conserved Non-Genic Sequences (CNGs). Science **302**:1033-1035.

Donoghue, P. C., and M. A. Purnell. 2005. Genome duplication, extinction and vertebrate evolution. Trends Ecol Evol **20**:312-319.

Force, A., M. Lynch, F. B. Pickett, A. Amores, Y. L. Yan, and J. Postlethwait. 1999. Preservation of duplicate genes by complementary, degenerative mutations. Genetics **151**:1531-1545.

Greer, J. M., J. Puetz, K. R. Thomas, and M. R. Capecchi. 2000. Maintenance of functional equivalence during paralogous Hox gene evolution. Nature **403**:661-665.

Gregory, T. R. 2002. Genome size and developmental complexity. Genetica **115**:131-146.

Halanych, K. M. 2004. The New View of Animal Phylogeny. Annual Review of Ecology, Evolution, and Systematics **35**:229-256.

He, X., and J. Zhang. 2005. Rapid subfunctionalization accompanied by prolonged and substantial neofunctionalization in duplicate gene evolution. Genetics **169**:1157-1164.

Hoekstra, H. E., and J. A. Coyne. 2007. The locus of evolution: evo devo and the genetics of adaptation. Evolution Int J Org Evolution **61**:995-1016.

Hubbard, T. J., B. L. Aken, S. Ayling, B. Ballester, K. Beal, E. Bragin, S. Brent, Y. Chen, P. Clapham, L. Clarke, G. Coates, S. Fairley, S. Fitzgerald, J. Fernandez-Banet, L. Gordon, S. Graf, S. Haider, M. Hammond, R. Holland, K. Howe, A. Jenkinson, N. Johnson, A. Kahari, D. Keefe, S. Keenan, R. Kinsella, F. Kokocinski, E. Kulesha, D. Lawson, I. Longden, K. Megy, P. Meidl, B. Overduin, A. Parker, B. Pritchard, D. Rios, M. Schuster, G. Slater, D. Smedley, W. Spooner, G. Spudich, S. Trevanion, A. Vilella, J. Vogel, S. White, S. Wilder, A. Zadissa, E. Birney, F. Cunningham, V. Curwen, R. Durbin, X. M. Fernandez-Suarez, J. Herrero, A. Kasprzyk, G. Proctor, J. Smith, S. Searle, and P. Flicek. 2008. Ensembl 2009. Nucleic Acids Res.

Hubbard, T. J., B. L. Aken, K. Beal, B. Ballester, M. Caccamo, Y. Chen, L. Clarke, G. Coates, F. Cunningham, T. Cutts, T. Down, S. C. Dyer, S. Fitzgerald, J. Fernandez-Banet, S. Graf, S. Haider, M. Hammond, J. Herrero, R. Holland, K. Howe, K. Howe, N. Johnson, A. Kahari, D. Keefe, F.

Kokocinski, E. Kulesha, D. Lawson, I. Longden, C. Melsopp, K. Megy, P. Meidl, B. Ouverdin, A. Parker, A. Prlic, S. Rice, D. Rios, M. Schuster, I. Sealy, J. Severin, G. Slater, D. Smedley, G. Spudich, S. Trevanion, A. Vilella, J. Vogel, S. White, M. Wood, T. Cox, V. Curwen, R. Durbin, X. M. Fernandez-Suarez, P. Flicek, A. Kasprzyk, G. Proctor, S. Searle, J. Smith, A. Ureta-Vidal, and E. Birney. 2007. Ensembl 2007. Nucleic Acids Res **35**:D610-617.

Huerta-Cepas, J., H. Dopazo, J. Dopazo, and T. Gabaldon. 2007. The human phylome. Genome Biol **8**:R109.

Hurles, M. 2004. Gene duplication: the genomic trade in spare parts. PLoS Biol **2**:E206.

ICGSC. 2004. Sequence and comparative analysis of the chicken genome provide unique perspectives on vertebrate evolution. Nature **432**:695-716.

Kondrashov, F. A., and A. S. Kondrashov. 2006. Role of selection in fixation of gene duplications. J Theor Biol **239**:141-151.

Li, L., C. J. Stoeckert, Jr., and D. S. Roos. 2003. OrthoMCL: identification of ortholog groups for eukaryotic genomes. Genome Res **13**:2178-2189.

Li, W.-H. 1997. Molecular evolution. Sinauer, Sunderland, MA.

Liolios, K., N. Tavernarakis, P. Hugenholtz, and N. C. Kyrpides. 2006. The Genomes On Line Database (GOLD) v.2: a monitor of genome projects worldwide. Nucleic Acids Res **34**:D332-334.

Long, M., E. Betran, K. Thornton, and W. Wang. 2003. The Origin of New Genes: Glimpses from the Young and Old. Nature Reviews Genetics **4**:865.

Lynch, M., and J. S. Conery. 2000. The evolutionary fate and consequences of duplicate genes. Science **290**:1151-1155.

Lynch, M., and J. S. Conery. 2003. The origins of genome complexity. Science **302**:1401-1404.

Lynch, M., and A. Force. 2000. The probability of duplicate gene preservation by subfunctionalization. Genetics **154**:459-473.

Martinez-Morales, J. R., T. Henrich, M. Ramialison, J. Wittbrodt, and J. R. Martinez-Morales. 2007. New genes in the evolution of the neural crest differentiation program. Genome Biol **8**:R36.

Milinkovitch, M. C., T. Gabaldon, and A. C. Tzika. submitted. 2× genomes—Depth does matter.

Milinkovitch, M. C., and A. Tzika. 2007. Escaping the mouse trap: the selection of new Evo-Devo model species. J Exp Zoolog B Mol Dev Evol **308B**:337-346.

O'Brien, K. P., M. Remm, and E. L. Sonnhammer. 2005. Inparanoid: a comprehensive database of eukaryotic orthologs. Nucleic Acids Res **33**:D476-480.

Ohno, S. 1970. Evolution by gene duplication. Springler Verlag, Heidelberg.

Rastogi, S., and D. A. Liberles. 2005. Subfunctionalization of duplicated genes as a transition state to neofunctionalization. BMC Evol Biol **5**:28.

Remm, M., C. E. Storm, and E. L. Sonnhammer. 2001. Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. J Mol Biol **314**:1041-1052.

Shiu, S. H., J. K. Byrnes, R. Pan, P. Zhang, and W. H. Li. 2006. Role of positive selection in the retention of duplicate genes in mammalian genomes. Proc Natl Acad Sci U S A **103**:2232-2236.

Springer, M. S., M. J. Stanhope, O. Madsen, and W. W. de Jong. 2004. Molecules consolidate the placental mammal tree. Trends Ecol Evol **19**:430-438.

Theissen, G. 2002. Secret life of genes. Nature **415**:741-741.

Tzika, A., R. Helaers, Y. Van de Peer, and M. C. Milinkovitch. 2008. MANTiS: a phylogenetic framework for multi-species genome comparisons. Bioinformatics **24**:151-157.

Van de Peer, Y. 2004. Computational approaches to unveiling ancient genome duplications. Nature Reviews Genetics **5**:752-763.

Vilella, A. J., J. Severin, A. Ureta-Vidal, R. Durbin, L. Heng, and E. Birney. 2008. EnsemblCompara GeneTrees: Analysis of complete, duplication aware phylogenetic trees in vertebrates. Genome Res.

2X GENOMES—DEPTH DOES MATTER

Akaike H: **A new look at the statistical model identification**. *IEEE Transactions on Automatic Control* 1974, **19**(6): 716–723.

Alexeyenko A, Tamas I, Liu G, Sonnhammer EL: **Automatic clustering of orthologs and inparalogs shared by multiple proteomes**. *Bioinformatics (Oxford, England)* 2006, **22**(14):e9-15.

Bashir A, Ye C, Price AL, Bafna V: **Orthologous repeats and mammalian phylogenetic inference**. *Genome Res* 2005, **15**(7):998-1006.

Benton MJ, Donoghue PC: **Paleontological evidence to date the tree of life**. *Mol Biol Evol* 2007, **24**(1):26-53.

Blomme T, Vandepoele K, De Bodt S, Simillion C, Maere S, Van de Peer Y: **The gain and loss of genes during 600 million years of vertebrate evolution**. *Genome biology* 2006, **7**(5):R43.

Dunn CW, Hejnol A, Matus DQ, Pang K, Browne WE, Smith SA, Seaver E, Rouse GW, Obst M, Edgecombe GD *et al*: **Broad phylogenomic sampling improves resolution of the animal tree of life**. *Nature* 2008, **452**(7188):745-749.

Edgar RC: **MUSCLE: a multiple sequence alignment method with reduced time and space complexity**. *BMC Bioinformatics* 2004, **5**:113.

Gabaldon T: **Large-scale assignment of orthology: back to phylogenetics?** *Genome biology* 2008, **9**(10):235.

Green P: **2x genomes--does depth matter?** *Genome Res* 2007, **17**(11):1547-1549.

Guindon S, Gascuel O: **A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood**. *Syst Biol* 2003, **52**(5):696-704.

Halanych KM: **The New View of Animal Phylogeny**. *Annual Review of Ecology, Evolution, and Systematics* 2004, **35**(1):229-256.

Hubbard TJ, Aken BL, Ayling S, Ballester B, Beal K, Bragin E, Brent S, Chen Y, Clapham P, Clarke L *et al*: **Ensembl 2009**. *Nucleic Acids Res* 2008.

Hubbard TJ, Aken BL, Beal K, Ballester B, Caccamo M, Chen Y, Clarke L, Coates G, Cunningham F, Cutts T *et al*: **Ensembl 2007**. *Nucleic Acids Res* 2007, **35**(Database issue):D610-617.

Huerta-Cepas J, Bueno A, Dopazo J, Gabaldon T: **PhylomeDB: a database for genome-wide collections of gene phylogenies**. *Nucleic Acids Res* 2008, **36**(Database issue):D491-496.

Huerta-Cepas J, Dopazo H, Dopazo J, Gabaldon T: **The human phylome**. *Genome biology* 2007, **8**(6):R109.Li L, Stoeckert CJ, Jr., Roos DS: **OrthoMCL: identification of ortholog groups for eukaryotic genomes**. *Genome Res* 2003, **13**(9):2178-2189.

Lander ES, Linton LM, Birren B, Nusbaum C, Zody MC, Baldwin J, Devon K, Dewar K, Doyle M, FitzHugh W *et al*: **Initial sequencing and analysis of the human genome**. *Nature* 2001, **409**(6822):860-921.

Liolios K, Tavernarakis N, Hugenholtz P, Kyrpides NC: **The Genomes On Line Database (GOLD) v.2: a monitor of genome projects worldwide**. *Nucleic Acids Res* 2006, **34**(Database issue):D332-334.

Milinkovitch MC, Tzika A: **Escaping the mouse trap: the selection of new Evo-Devo model species**. *Journal of experimental zoology Part B* 2007, **308B**(4):337-346.

Smith TF, Waterman MS: **Identification of common molecular subsequences**. *J Mol Biol* 1981, **147**(1):195-197.

Springer MS, Stanhope MJ, Madsen O, de Jong WW: **Molecules consolidate the placental mammal tree**. *Trends in ecology & evolution (Personal edition)* 2004, **19**(8):430-438.

Tzika A, Helaers R, Van de Peer Y, Milinkovitch MC: **MANTiS: a phylogenetic framework for multi-species genome comparisons**. *Bioinformatics (Oxford, England)* 2008, **24**(2):151-157.

Venter JC, Adams MD, Myers EW, Li PW, Mural RJ, Sutton GG, Smith HO, Yandell M, Evans CA, Holt RA *et al*: **The sequence of the human genome**. *Science* 2001, **291**(5507):1304-1351.

Vilella AJ, Severin J, Ureta-Vidal A, Heng L, Durbin R, Birney E: **EnsemblCompara GeneTrees: Complete, duplication-aware phylogenetic trees in vertebrates**. *Genome Res* 2009, **19**(2):327-335.

Zmasek C, Eddy S: **A simple algorithm to infer gene duplication and speciation events on a gene tree**. *Bioinformatics (Oxford, England)* 2001, **17**:821–828.

HISTORICAL CONSTRAINTS ON VERTEBRATE GENOME EVOLUTION

Alexeyenko, A., I. Tamas, G. Liu, and E. L. Sonnhammer. 2006. Automatic clustering of orthologs and inparalogs shared by multiple proteomes. Bioinformatics **22**:e9-15.

Bashir, A., C. Ye, A. L. Price, and V. Bafna. 2005. Orthologous repeats and mammalian phylogenetic inference. Genome Res. **15**:998-1006.

Bejerano, G., M. Pheasant, I. Makunin, S. Stephen, W. J. Kent, J. S. Mattick, and D. Haussler. 2004. Ultraconserved Elements in the Human Genome. Science **304**:1321-1325.

Carroll, S. B., J. K. Grenier, and S. D. Weatherbee. 2005. From DNA to diversity, molecular genetics and the evolution of animal design. Blackwell publishing, Malden.

Clark, A. G., S. Glanowski, R. Nielsen, P. D. Thomas, A. Kejariwal, M. A. Todd, D. M. Tanenbaum, D. Civello, F. Lu, B. Murphy, S. Ferriera, G. Wang, X. Zheng, T. J. White, J. J. Sninsky, M. D. Adams, and M. Cargill. 2003. Inferring Nonneutral Evolution from Human-Chimp-Mouse Orthologous Gene Trios. Science **302**:1960-1963.

Dermitzakis, E. T., A. Reymond, N. Scamuffa, C. Ucla, E. Kirkness, C. Rossier, and S. E. Antonarakis. 2003. Evolutionary Discrimination of Mammalian Conserved Non-Genic Sequences (CNGs). Science **302**:1033-1035.

Force, A., M. Lynch, F. B. Pickett, A. Amores, Y. L. Yan, and J. Postlethwait. 1999. Preservation of duplicate genes by complementary, degenerative mutations. Genetics **151**:1531-1545.

Gabaldon, T. 2008. Large-scale assignment of orthology: back to phylogenetics? Genome Biol **9**:235.

Greer, J. M., J. Puetz, K. R. Thomas, and M. R. Capecchi. 2000. Maintenance of functional equivalence during paralogous Hox gene evolution. Nature **403**:661-665.

Halanych, K. M. 2004. The New View of Animal Phylogeny. Annual Review of Ecology, Evolution, and Systematics **35**:229-256.

Hoekstra, H. E., and J. A. Coyne. 2007. The locus of evolution: evo devo and the genetics of adaptation. Evolution Int J Org Evolution **61**:995-1016.

Hubbard, T. J., B. L. Aken, S. Ayling, B. Ballester, K. Beal, E. Bragin, S. Brent, Y. Chen, P. Clapham, L. Clarke, G. Coates, S. Fairley, S. Fitzgerald, J. Fernandez-Banet, L. Gordon, S. Graf, S. Haider, M. Hammond, R. Holland, K. Howe, A. Jenkinson, N. Johnson, A. Kahari, D. Keefe, S. Keenan, R. Kinsella, F. Kokocinski, E. Kulesha, D. Lawson, I. Longden, K. Megy, P. Meidl, B. Overduin, A. Parker, B. Pritchard, D. Rios, M. Schuster, G. Slater, D. Smedley, W. Spooner, G. Spudich, S. Trevanion, A. Vilella, J. Vogel, S. White, S. Wilder, A. Zadissa, E. Birney, F.

Cunningham, V. Curwen, R. Durbin, X. M. Fernandez-Suarez, J. Herrero, A. Kasprzyk, G. Proctor, J. Smith, S. Searle, and P. Flicek. 2008. Ensembl 2009. Nucleic Acids Res.

Hubbard, T. J., B. L. Aken, K. Beal, B. Ballester, M. Caccamo, Y. Chen, L. Clarke, G. Coates, F. Cunningham, T. Cutts, T. Down, S. C. Dyer, S. Fitzgerald, J. Fernandez-Banet, S. Graf, S. Haider, M. Hammond, J. Herrero, R. Holland, K. Howe, K. Howe, N. Johnson, A. Kahari, D. Keefe, F. Kokocinski, E. Kulesha, D. Lawson, I. Longden, C. Melsopp, K. Megy, P. Meidl, B. Ouverdin, A. Parker, A. Prlic, S. Rice, D. Rios, M. Schuster, I. Sealy, J. Severin, G. Slater, D. Smedley, G. Spudich, S. Trevanion, A. Vilella, J. Vogel, S. White, M. Wood, T. Cox, V. Curwen, R. Durbin, X. M. Fernandez-Suarez, P. Flicek, A. Kasprzyk, G. Proctor, S. Searle, J. Smith, A. Ureta-Vidal, and E. Birney. 2007. Ensembl 2007. Nucleic Acids Res **35**:D610-617.

Huerta-Cepas, J., A. Bueno, J. Dopazo, and T. Gabaldon. 2008. PhylomeDB: a database for genome-wide collections of gene phylogenies. Nucleic Acids Res **36**:D491-496.

Huerta-Cepas, J., H. Dopazo, J. Dopazo, and T. Gabaldon. 2007. The human phylome. Genome Biol **8**:R109.

Kelso, J., J. Visagie, G. Theiler, A. Christoffels, S. Bardien, D. Smedley, D. Otgaar, G. Greyling, C. V. Jongeneel, M. I. McCarthy, T. Hide, and W. Hide. 2003. eVOC: A Controlled Vocabulary for Unifying Gene Expression Data. Genome Res. %R 10.1101/gr.985203 **13**:1222-1230.

Li, L., C. J. Stoeckert, Jr., and D. S. Roos. 2003. OrthoMCL: identification of ortholog groups for eukaryotic genomes. Genome Res **13**:2178-2189.

Li, W.-H. 1997. Molecular evolution. Sinauer, Sunderland, MA.

Lynch, M., and J. S. Conery. 2000. The evolutionary fate and consequences of duplicate genes. Science **290**:1151-1155.

Lynch, M., and A. Force. 2000. The probability of duplicate gene preservation by subfunctionalization. Genetics **154**:459-473.

Pao, S.-Y., W.-L. Lin, and M.-J. Hwang. 2006. In silico identification and comparative analysis of differentially expressed genes in human and mouse tissues. Bmc Genomics **7**:1-11.

Roux, J., and M. Robinson-Rechavi. 2008. Developmental constraints on vertebrate genome evolution. PLoS Genet **4**:e1000311.

Springer, M. S., M. J. Stanhope, O. Madsen, and W. W. de Jong. 2004. Molecules consolidate the placental mammal tree. Trends Ecol Evol **19**:430-438.

Su, A. I., M. P. Cooke, K. A. Ching, Y. Hakak, J. R. Walker, T. Wiltshire, A. P. Orth, R. G. Vega, L. M. Sapinoso, A. Moqrich, A. Patapoutian, G. M. Hampton, P. G. Schultz, and J. B. Hogenesch. 2002. Large-scale analysis of the human and mouse transcriptomes. PNAS **99**:4465-4470.

Theissen, G. 2002. Secret life of genes. Nature **415**:741-741.

Tzika, A., R. Helaers, Y. Van de Peer, and M. C. Milinkovitch. 2008. MANTiS: a phylogenetic framework for multi-species genome comparisons. Bioinformatics **24**:151-157.

Valentine, J. W., A. G. Collins, and C. P. Meyer. 1994. Morphological Complexity Increase in Metazoans. Paleobiology **20**:131-142.

Vilella, A. J., J. Severin, A. Ureta-Vidal, L. Heng, R. Durbin, and E. Birney. 2009. EnsemblCompara GeneTrees: Complete, duplication-aware phylogenetic trees in vertebrates. Genome Res **19**:327-335.